

2011.05

# PROGRAMMER

# 程序员

## 云计算应用之路

2011年，中国云计算开启落地之旅，究竟多少风光和险滩？  
12家代表性企业云实践分享 • 7位云计算创业者深入访谈  
图解全球云计算大势 • 展望中国云计算未来之路

26 苹果 App Store 引发的  
**悲喜狂欢**

35  **Color的  
3个问题**

48 **2011中国云计算  
风云展望**

66 **hulu**  
推荐系统在视频网站中的应用

90 敏捷大师  
**Fred George**  
我倡导无政府主义编程

93 **facebook**  
工程管理揭秘之  
内部晋升

100 一分钟先生  
**软件研发中的冲突  
及解决之道**

128  **Mac OS X  
背后的故事**

苹果公司一度成为漏水的船，Ellen Hancock  
力挽狂澜，为Mac OS X系统的诞生奠定基础。

ISSN 1672-3252



邮发代号：2-665 定价：15元

www.programmer.com.cn www.csdn.net

# Contents

## 固定专栏

7	外刊速递
10	简讯
12	新产品新工具
14	程序天下事
120	Geek
122	新书上架
134	10年前
136	漫画与幽默
131	名人堂

## 报道

20	让天下没有难卖的软件 ——实记CSGS2011
22	春天里的平板电脑和数据中心 ——2011英特尔信息技术峰会有感

## 移动专区

24	热点资讯TOP10
26	苹果APP Store引发的悲喜狂欢 近日苹果预调整App Store的排名算法以及限制非自然下载，在业界掀起轩然大波，它将产生哪些影响，独立开发者如何应对？请听本文作者的分析。

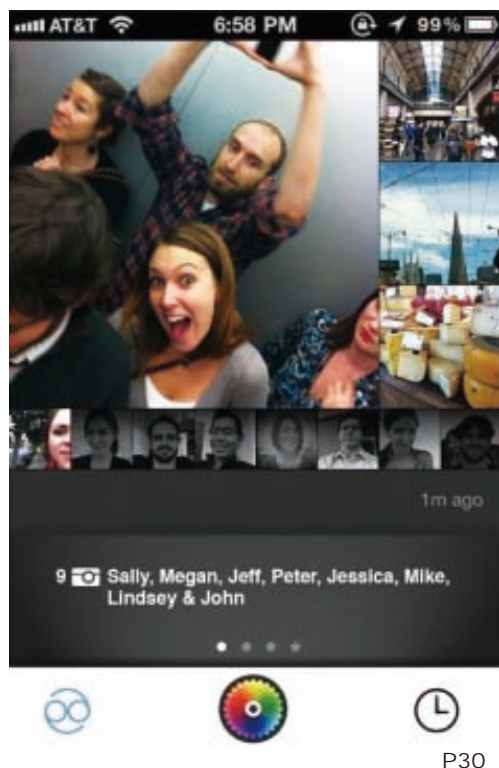
## 专题：解读Color

移动应用新星Color正式发布前竟获4000万美元投资，中国版本的Color也迅速应运而生，究竟内含何种奥妙，移动社交呈现何种变局，专家向您详细解读。

30	Color背后：社交互动新模式
32	中国版Color更给力 ——专访“推图”创始人陈仲军
35	关于Color的三个问题

## 专访

36	移动新媒体融合之道 ——专访21世纪新媒体CTO韩磊
----	-------------------------------



# Contents

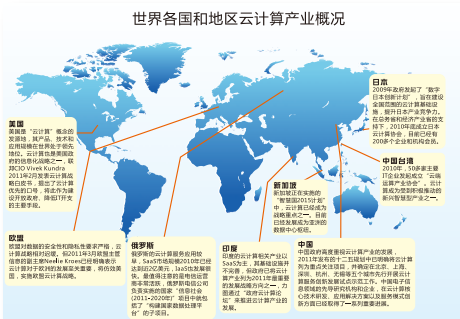
39

从应用创作者到产品总监  
——专访微软MVP邹建峰

技术

40

Android操作系统移植经验大分享（上）  
本文从技术角度出发，搭配众多实例对Android操作系统的移植进行了详尽介绍。



P46

封面报道：云计算应用之路

44

开篇

46

云计算大势图

48

大风起兮云飞扬  
——记2011年的中国云计算

53

亲密接触中国SaaS应用

58

看云卷云舒  
——各公司云计算之路漫谈

62

大规模图处理系统——Pregel

66

推荐系统在视频网站中的应用

70

Amazon EC2：性能测试的云中基石

74

基于分布式Key-Value DB和分布式文件系统的海量云存储系统构建

78

淘宝Oceanbase云存储系统实践

82

有向无环图之Dryad

87

云计算创业者说

工程管理

90

我倡导无政府主义编程  
——Fred George访谈录

93

Facebook的内部晋升

对话CTO

96

做可延续的运营平台  
风灵创景CEO张磊认为一个优秀的操作系统不仅仅是孤零零的与硬件配合，而应该是一个帮助用户提升整体性能的运营平台。



P90

# Contents

主管：中国社会科学院  
主办：中国社会科学院文献信息中心  
出版：《程序员》杂志社  
网址：http://www.programmer.com.cn  
国际刊号：ISSN 1672-3252  
国内刊号：CN11-5038/G2  
邮发代号：2-665  
广告经营许可证号：京东工商广字0188号

总编：黄长著 Editor-in-chief: Huang Changzhu  
社长/常务副总编：张悦校 President: Zhang Yue Xiao  
副社长：蒋涛 Vice President: Jiang Tao  
编委会：黄长著 张悦校 陈洋彬 蒋涛 曾登高 刘江  
Editorial Member: Huang Changzhu Zhang Yue Xiao Chen Yangbin  
Jiang Tao Zeng Denggao Liu Jiang

执行主编：孟迎霞 Executive Editor-in-chief: Meng Yingxia  
编辑部主任：常政 Director: Chang Zheng  
责任编辑：董世晓 高松 郑柯  
Editors: Dong Shixiao Gao Song Zheng Ke  
特邀编辑：方梁 高昂 赵健平 吕娜 卢鹤翔  
Contributing Editors: Fang Liang Gao Aang Zhao Jianping Lv Na

Lu Dongxiang  
美术设计：纪明超 Art Designer: Ji Mingchao  
美术编辑：林象海 Art Editor: Lin Xianghai  
流程编辑：陈秋歌 Coordinator: Chen Qiuge  
Tel: 010-64351458  
Email: editor@csdn.net

发行部 Distribution Dept.010-64351431  
Email: sales@csdn.net

广告总代理：北京创新乐知广告有限公司  
Sole Advertising Agency: Beijing CSDN Co.,Ltd  
Tel: 010-64376055  
Email: ad@csdn.net  
Marketing Dept: 010-51661202 (ext 149)  
Email: market@csdn.net

读者服务部  
Readers service Dept.  
网上订购：http://dingyue.programmer.com.cn/  
读者信箱：reader@csdn.net  
地址：北京市朝阳区广顺北大街33号院1号楼福码大厦B座12层  
Address: B-12th Floor Fairmont Tower NO.33 Guangshun North street,  
Chaoyang District, Beijing  
邮政编码：100102  
电话：010-64351436  
传真：010-64348545

法律顾问：北京中润律师事务所 王杰  
Law Consultant: Beijing Hengsheng Lawyer Firm  
印刷：北京盛通印刷股份有限公司  
Print: Beijing Shengtong Printing Co., Ltd.  
出版日期：每月1日  
Publication Date: the first day per month  
零售价：RMB 15.00元 新台币 390元 HK \$ 35.00 (港、澳)  
US \$ 9.00 (海外)  
Retail Price: RMB 15, NT\$390, HK \$ 35.00, US \$ 9.00

本刊文章版权所有 未经许可不得转载发现装订错误或缺页，请将杂志寄回本刊读者服务部，即可得到调换。

## 98

### 一分钟先生

## 100

### TUP专栏

## 102

## 106

## 107

### 算法

## 110

### 并行编程

## 114

### 调试之剑

## 116

### 程序人生

## 124

### 软件春秋

## 128

### 观点

## 132

## 133

#### 用熟悉的技术做产品

人人网技术总监黄晶认为，一个软件系统是不是健康，与整个系统架构的设计有着重要关系，因此在技术选型上，最好是应用自己和团队熟悉的技术。

#### 软件开发中的冲突及解决之道

#### 产品管理与软件开发的关系

本文是前eBay高级副总裁Marty Cagan回顾自己二十多年来从事软件产品管理工作的总结和经验分享，谈到了产品管理与软件开发的关系，以及软件开发人员如何转型做产品管理。

#### 没有永恒的产品经理

#### 架构师接龙：去哪儿网孙立 vs. 飞信孙朝晖

如何在架构层面，提高开发人员开发效率？如何搭建更加有效的测试环境？如何在架构层面帮助快速定位故障的原因？精彩问答，尽在本期架构师接龙。

#### 揭秘热文排序算法

——SmartHot

本文结合我烧网实战，详细介绍了它的SmartHot热文排序算法。

#### 并行编程会像当年的Java一样流行

#### 解救即将被断网的系统

——调试补丁安装失败（上）

#### 亚嵌李明：一生的学习

#### Mac OS X背后的故事（一）： 力挽狂澜的Ellen Hancock

#### 新泡沫 新规则

#### 软件架构与程序员的关系



# Communications of ACM

2011.04



## 《蔓延到社交媒体的平台大战》

过去半个世纪里，我们目睹了计算和沟通平台的巨大发展。特别是在过去几年里，我们看到了社交网络的兴起，从MySpace到Facebook、LinkedIn、Twitter、Foursquare，还有Groupon（他们最近拒绝了Google出价65亿美元的收购）。

社交媒体网络属于不同程度上的新型平台，它们促进了沟通，或者提供了发送电子邮件、分享文件的新系统。对于这些平台如何与其他平台（例如个人计算平台——微软公司的Windows和苹果的Mac系统、通用移动通讯平台——RIM的Blackberry、Nokia的Symbian、Google的Android、微软的Windows、传统的电子邮箱平台——微软的Outlook和Google的Gmail，互联网搜索平台——Google、Firefox和Bing，还有文件共享平台——Flickr等等）争夺注意力和收益，人们了解的比较少。《蔓延到社交媒体的平台大战》一文中，作者认为事实上，在社交媒体领域，一场平台大战已经打响。

如何在战争中获胜或者守住自己的领地，之前的平台大战为我们提供了很好的借鉴。但是，在涉及社交网络时，要牢记两点：一是我们现在看到的成功者，例如Facebook还有Groupon，都是大浪淘沙后的幸存者，大部分的社交媒体会销声匿迹或者最终默默无闻，它们无法依赖传统的广告商业模式生存。和搜索引擎中的广告不同，社交网络中的广告会被认为是干扰和对隐私的侵犯。虽然Groupon创造了新的商业模式，并且最终获得了成功。但遗憾的是，这种模式很容易被“山寨”。它需要建立一个具有防御能力的平台和更广泛的生态系统，来保持它的领先地位。二是在平台大战中，最终胜利的是最好的平台，而不是最好的商品（或服务）。VHS录像系统战胜Betamax录像系统、MS-DOS和微软的PC战胜MAC机、Intel的x86微处

理器战胜非兼容的芯片、甚至芭比娃娃战胜后来者贝兹娃娃的事例都可以说明这一点。

最后，作者总结了获胜平台的几个特征：能够产生强大的网络效应；利用差异化战略和市场细分，让竞争对手没有机会分裂市场；让用户或者生态系统中的合作者（主要是互补的应用产品和服务的开发者，以及广告商）很难使用多个平台，或者要为之付出很大代价。

## 《视觉传播的设计原则》

通过图解、示意图、图表、照片、视频和动画进行的视觉传播（Visual Communication），是探索概念和传播信息的基础。有效的可视化调动了人类处理视觉信息的能力，可以增进理解、记忆和推理。因此可视化可以帮助分析师快速地找到隐藏在大量数据中的模式，并帮助受众快速地理解复杂的概念。

最好的可视化效果是由熟练的设计师人工设计出来的，但是，即使有计算机的帮助，手工设计可视化效果仍然是件费时费力的事情。而且全世界制造出的新数据量成指数级的增长。但是对这些数据的收集和储存，并不能带来什么价值，我们必须理解这些数据，从中挖掘其真正的价值，并利用它们做出决策。

很多需要利用可视化技术的分析师和终端用户都不是受过专业训练的设计师，这里给出几个原则，可以帮助非设计人员创建有效的可视化效果。《视觉传播的设计原则》一文中，作者开发了一种通用的三步法，来建立某个特定领域内的可视化设计系统：

第一步：找出设计原则。为了提取出特定领域的设计原则，并将其系统化，可以采用以下3个策略：分析这个领域最棒的人工可视化设计；检查之前对可视化理解认知的研究；进行新的用户研究，以考察视觉技术如何影响人们

的理解认知。

第二步：将设计原则实例化。设计原则可以分为两大类：设计规则和评价指标。设计规则可以区分哪些是有效的设计，哪些是不可行的设计；评价指标将可视化的效果量化。有了设计规则和评价指标，利用程序技术可以建立一个自动可视化设计系统。

第三步：评估设计原则。在这一步，对自动设计系统产生的可视化效果的可用性进行衡量。衡量指标包括用户的反馈（以定性的访谈和定量的使用统计两种方式）及用户研究（检查可视化可以在多大程度上改善信息的处理、传播和决策）。

《实现IPv6突破的制胜策略》

1973年，人们开始设计TCP/IP协议；1977年，TCP/IP的设计者们开始考虑这个问题：“互联网需要多大的地址空间？”对于这个问题，互联网工程师以及科学家们进行了将近一年时间的辩论，最后确定使用32个比特位来代表互联网地址，在当时的情况下，43亿个可用的地址似乎已经足够。但日前，负责分配全球IP地址的ICANN宣布最后5

个A级IPv4地址块已经分配完毕。我们已经用尽了32位的IP地址（IPv4），开始向IPv6格式的128位IP地址迁移。

因为纯IPv4没有足够的地址空间指向纯IPv6的目标，所以IPv6格式无法向下兼容IPv4。因此在这两种协议共存的时期，有必要采用双重架构（dual-stack）设计，能够让主机同时使用这两种协议。但最终，纯IPv6主机将会取代IPv4。向IPv6的过渡将是互联网架构的一次最重大的变革，要保证用户、服务器、互联网服务和访问提供商能够同时使用旧协议和新协议，需要很多人投入很多精力。《实现IPv6突破的制胜策略》一文中，Thomas Limoncelli设想了要达到这个目标可以采取的几个步骤，他通过一个失败的和两个成功的策略，说明了以下两点。

一次将所有相关设施更换成IPv6是件很疯狂的事情，而且会遭到管理者的回绝。因为当下，进行如此彻底的变革不能带来明显的商业价值。

遵循由外而内的原则。使用负载均衡器进行IPv6到IPv4的转换，使你现在能够为外部的消费者提供IPv6服务，为你带来“快速优势”以支撑未来的项目，并且让你可以控制变革的步伐；提出使用IPv6能够带来高价值的理由——“闪光点”，更容易获得管理人员的首肯。



Wired  
2011.04

《Kickstarter如何成为大胆的技术原型和创意产品的实验场》

构思出一个优秀的创意，世界上的机遇便会纷至沓来。这是个令人愉快的观念，鼓舞了不少的发明家，但实际上，从创意出炉到最终成为产品，是个艰难而漫长的过程。而现在，Kickstarter可以帮助发明家迅速地将自己的创意变成现实。

《Kickstarter如何成为大胆的技术原型和创意产品的实验场》介绍了Kickstarter的诞生和发展情况。这家网站创建于2009年，以众包（crowdsourcing）的方式为不同寻常的艺术项目提供资助。已经有超过14000人在Kickstarter上发布了项目，超过40万人为这些项目提供了资助，资助总额超过3500万美元。每天新启动的项目有80个，每周的资助额有100万美元。该网站开创了一种前所未有的资金筹集渠道。

最初，网站只是为创意艺术项目筹集资金。但是后来的投资者看到了这个网站更大的商业潜力，因为它可以建立与受众之间更紧密的联系，也展现出了很大的市场营销潜力。现在，即使是那些本不需要通过Kickstarter筹集资金的人们也开始把目光投向这个网站。工业设计师Scott Wilson为了避开“大公司的政治和优柔寡断”，决定使用Kickstarter来开启自己的新项目，他的筹集目标是15000美元，30天之后，他得到了13512人总额941778美元的资助，这是迄今为止，Kickstarter上筹集到最多资金的项目。

Kickstarter中存在着巨大的商机，但至少在目前，该网站的创始人有意地回避这些机会。因为随着募集金额的增大，要面临到一些问题：得到资助却没有完成项目；募集的资金额度超出实际所需；欺诈行为。他们计划建立信用系统以解决这些问题。

虽然Kickstarter一直试图保持低调，但这种情况不会持续很久。因为就像当初的Twitter一样，最有趣的公司可以激发出很多新奇的行为，网站的创建者们永远料不到人们会如何使用这些服务。

## 《访谈：开源电子先驱Limor Fried谈DIY革命》

Limor Fried可以称作是最一流的极客：她在麻省理工拿到了电子工程的学位，用Altoid薄荷糖锡罐做过几项天才发明，还对传奇的Roland TB-303合成器进行过逆向工程。现在她运营着Adafruit Industries公司，为目前不断增加的DIY爱好者提供开源电子工具和组件。

本期的《连线》杂志就DIY革命对她进行了专访，记录在《访谈：开源电子先驱Limor Fried谈DIY革命》一文中。

在问及21世纪制造者的特征时，她表示：与过去相比，现在记录项目和分享项目都变得更简便了。现在人们可以用视频来记录制作过程，而且有很多网站可以让人们分享自己的项目，形成了更大的社团，可以进行合作。她认为这些项目可以用数字形式储存并在线传播，起到了催化的作用。借助这些数字化的文件，人们可以直接利用现有的技术和借鉴别人的工作。

如今DIY的风行与在线销售零件的网站有很大关系。几年前，要获得DIY所需的零件还不那么容易，但现在什么都可以很方便地从网上获得。DIY是一种爱好，但是就像上世纪六七十年代在车库里制作计算机的人们最终影响了计算机的发展一样，DIY爱好者可能是某项技术的先驱，有可能会创造出某种工业模型。

操作一台机械很简单，但是更重要的是设计、创新和

工程学。制造者运动有望能锻炼人们的这些能力，推动制造业的发展。DIY革命会让人们逐渐地接受自己创造、修改事物的观念。它影响到的不仅仅是工程人员和科学家，还有艺术家以及普通人。对于消费者来说，一方面可以获得更加个性化的产品；另一方面，消费者可以选择的空间也更大了。

在谈到她创办的Adafruit Industries公司时，她说公司的目的并不是为了盈利，而是为了而是为了教授电子学知识和研究工程学，并且探索有趣的有创造性的项目。她认为这是一种科学工作，而不是为了某种具体的目的。

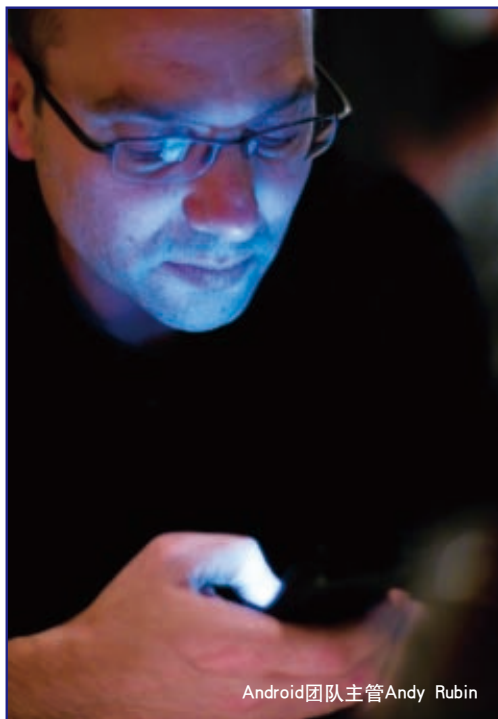
## 《Larry Page希望谷歌回归创业公司》

2001年，谷歌的创始人Larry Page和Sergey Brin采纳了投资人John Doerr的建议，接受Eric Schmidt成为公司的CEO，取代Larry Page。在Schmidt手下，谷歌成长为全球第3大技术公司，10年间其收益从1亿增长到今天的300亿。10年之后，Larry Page重新成为公司CEO。对公众来说，也许不乐见这次的重大变化——对大部分来说，他只是古怪的“谷歌兄弟”之一，但Page是特立独行的，他有潜力产生Bill Gates和Steve Jobs那样的影响力。没有人能比他更好地代表谷歌的野心、价值观和世界观。同时，他也是古怪、自大和神秘的。在他的领导下，这家公司的行为将更加难以预测。

2004年，谷歌向证券交易委员会提交的拟上市文件中，包含了Page给未来股东的一封信。其中，他宣布“谷歌不是一家传统意义上的公司，我们也无意成为这样的公司。”在之后的几年里，谷歌遵守了这个承诺。但是在“三驾马车”中，Schmidt用更多传统的做法平衡两位创始人特立独行的冲动。现在，Page执掌大局，没有人知道如何能维持这种微妙的平衡，或者说是否还能维持这种平衡。

Page野心极大，喜欢大而不当的想法。他被任命来复兴谷歌，从某些方面来说，他是执行这一任务的最佳人选。对于目前巨大的谷歌公司来说，官僚主义阻碍了进步。《Larry Page希望谷歌回归创业公司》一文认为，Page面对的是一项不可能的任务：要让一家拥有24000名员工的公司像一个创业公司那样运转。能不能完成这项任务，就要看Page是否已经具备了履行CEO常规职责的忍耐力、意愿和魅力，并且同时还能保留自己的独特个性。（感谢译者卞斌支持）





Android团队主管Andy Rubin

## Android的前途

Novell宣布发布Mono for Android; 同时, AMD开始招兵买马, 进军Android平台, 大家纷纷为Android做嫁衣裳。但Android 3.0 暂不开源的决定, 也让人心生嘀咕。

对此, Andy Rubin撰文, 重申了Android的开源一定会坚持下去, Android 3.0源码一定会放出。此文发布的同时, AOSP也开始释出部分Android 3.0代码, 虽然还没有Framework、Apps和Dalvik的3.0源码放出, 但确信这只是时间问题。目前Android团队应该是发现了Android 3.0的诸多问题, 在修复的同时整合成为手机、平板甚至电视的统一平台。正如著名自制ROM开发者Cyanogen提到的, “这不是坏事”。

Facebook真的即将进入中国, 协议正式签署, 国内某网站是落地合作方, 采取另建新站的落地模式, 完全与Facehook.com打通不可能。未来会“非死不可”吗?

——Facebook入华风声又起, DCCI互联网数据中心创始人胡延平发表评论。

如果从以前的几次技术革命来看, 中国往往是跟着国外的脚步, 学人家的东西。但中国在这次云计算的技术变革上会有突破性的机会。

——VMware中国研发中心总经理李严冰谈云计算发展。

App Store中99%的用户其实并没有兴致将这些无聊的游戏玩到头。因为绝大部分游戏制作得都很粗糙, 也没经过任何游戏测试, 这些游戏开发商甚至一点都不想认真把游戏做好。

——前苹果游戏设计师Graeme Devine指出iPhone开发者的错误。

## Twitter与搜索

4月初, Twitter工程师博客发文称: “用名为Blender的Java服务器取代了原先的RoR, 这一调整令搜索速度提升了三倍。”今年年初, Bing首席科学家Alek Kolcz跳槽至Twitter, 由此可见, 他的工作颇有成效。

之前, Twitter多与其他搜索巨头合作。Bing曾宣布包含Twitter信息。Google与Twitter协议, 可以访问其全部数据流。如今Twitter在做的就是不假他人之手, 以期获得更大收益。

Twitter搜索的重要性也在提升。4月19日, Twitter斥资5000万美元收购第三方桌面客户端TweetDeck, 意在保护自己的数据, 扼杀潜在对手。由此, Twitter的正反手初露锋芒。



## 360和金山安全

4月7日, 奇虎360发布浏览器应用开发平台。据周鸿祎说, “360是一张坚持开放平台的白纸”, 而360的开放策略已经陆续布局在浏览器、团购、页游、桌面等方面。这说明360的重心正在从装机量转入盈利。在一个巨大的流量体系已经建立和形成后, 360具备了从容强化盈利模式的资源保障。现在是需要考虑给投资者提供好的盈利答卷的时候了。

4月初, 金山安全低调收购驱动之家, 这被外界认为是继360收购鲁大师之后的应对之举, 今后金山还将深化卫士类安全辅助工具的产品内涵, 细化领域竞争, 如果不出所料, 金山卫士可能会加入驱动查找和硬件诊断功能。

## LinkedIn和Facebook

4月初, LinkedIn正式宣布全面开放开发者平台, 支持OAuth 2.0和多种API服务, 并推出了一系列插件。这与2000年, Facebook推出的“Open Graph”平台非常相似: 网站ID登录第三方网站, 与其他网站分享用户资料, “分享”按钮将内容发送至联系人; “推荐”按钮将网站或服务推荐给联系人, 唯一的区别是, LinkedIn没有将这个平台命名为“Open Graph”。

LinkedIn是目前全球最大的职业社交网络, 风头正劲, 不输于Facebook, 此举无疑直接应对Facebook, 争夺人气, 也让用户思考, 以后印到名片上的Profile, 是LinkedIn还是Facebook的呢?

苹果必须解释在何地、多长时间, 以及因何目的而保存这些数据, 以及有什么人可以获取这些数据, 提供了什么安全措施防止未经授权的读取等。

——德国要求苹果解释iPhone存储用户数据行为, 德国消费者保护部发言人Holger Eichele批评道。

我一直在保持公司组织结构上层体系的最小化。当有一个员工说出“晋升我吧”, 我不得不重新评估公司的组织结构。

——37signals创始人之一Jason Fried在一篇博文中谈如何管理公司、打造团队。

传统的DBMS在云计算环境中依然有立足之地

——Yahoo!云计算首席科学家Ramakrishnan评论云计算对传统数据库的影响。

移动正在重返专利性质。让人担心的是我们又一次粉碎网络, 破坏了它。不论是开发者还是用户都被迫作出选择。

——Mozilla CEO Gary Kovacs谈移动领域的现状。

# CSDN最受关注文章

2011年3月26日~4月26日

## 01 腾讯与新浪微博开放平台之比较

邵宝麟

创业者最为关心开放平台的API。从API多样性、数据完整性、请求限制三个维度,比较腾讯和新浪的微博平台,可以看出,新浪微博是国内目前最接近Twitter的微博平台,而腾讯微博目前只支持基础应用,其接口暂时还不能满足开发者的复杂应用。

## 02 比Kinect更强的视频追踪算法

boycott

在英国念书的捷克学生Zdenek Kalal,发明了一套神奇的精确定位系统: Predator, 几乎可以实现“Minority Report”般的人机界面。如同微软Xbox的Kinect一样,但这个效果更好,不用手就能简单地操控计算机,这项技术应用前景广泛。萨里大学网站免费提供程序的代码。

## 03 Google算法改变影响互联网

36氪

最近SEOBook绘制了一幅有趣的图,该图详细描绘了Google搜索算法的微妙改变会给互联网带来多大的影响。甚至随便更改几个数字都有可能逼迫一家网络公司关门歇业, Mahalo就是Google最近算法改变后的受害者,该公司因搜索流量的减少,不得不裁掉10%的员工。

## 04 人人网申请纽交所上市

陈秋歌

北京时间4月16日,人人网运营方人人公司向美国证券交易委员会(SEC)提交了招股说明书,正式启动首次公开招股(IPO)。同时在招股书中,人人公司提到了知识产权诉讼问题,承认游戏等内容可能侵权,同时可能遭到SNS的相关专利诉讼。

## 05 盖茨是乘人之危的小人?

孙行之

Paul Allen, 微软公司的联合创始人,出版了自传Idea Man。在Allen笔下,比尔·盖茨是盘算着窃取朋友劳动果实和财富的阴险小人,但公众似乎并不关心富豪之间的私人恩怨与是是非非。而盖茨在成为世界首富之后,宣布捐献了自己的全部财产。

## 06 2011年美国最佳工作排名

Andrew Strieber

美国招聘网站CareerCast.com发布了2011年最佳和最差工作排行榜,在多达200个工作榜单上,软件工程师高居榜首。事实上,软件工程师今年的压力排名已提高了10个等级,从第25位上升至第15位。



## 07 Google到底有多大?

Pingdom

据StatCounter统计, Google已经占有桌面搜索引擎90%的市场份额,手机搜索,则完全由Google统治,市场份额达到惊人的97%! 大多数智能手机默认搜索引擎就是Google。再加上其他服务(YouTube、Blogger等), Google全部用户群应该在10亿到15亿之间。

## 08 局域网战胜互联网

月光博客

2006年4月, 李彦宏预言, “5年以后, Google和百度之间的关系是百度一枝独秀,从市场份额来说,大家很难看到Google了。”如今,这个预言在某种程度上应验了。如果Google没有被屏蔽,局面是否还是今天这样? 劣币驱逐良币,是这个时代必须面对的问题。

## 09 华为发布2010年年报

李志宇

4月18日, 华为在官方网站发布了2010年年报。其中, 首次披露了所有董事会、监事会成员名单、简历和照片。此外, 还第一次向公众披露公司面向未来的组织架构, 并且第一次披露分部(区域、业务)财务数据, 其中海外收入超过六成。

## 10 硅谷第二代创新者涌现

张楠

名为“High School Memories”的Facebook应用,吸引了越来越多的用户。该应用主要帮助用户分享学生时代的记忆,它的开发者Cyrus Pishavar年仅13岁,尚未达到读高中的年龄。事实上,硅谷的许多年轻人选择在业余时间自行开发应用,甚至建立自己的公司。

## 2011年技术投资者排行榜(福布斯)

投资者 投资公司 投资对象

榜单排名的主要考虑因素是过去五年投资者通过IPO(首次公开募股)和并购交易获得的投资回报率,同时还利用了一系列其他因素,最后由一个专家小组确定最终的名单。

## 01 Jim Breyer

Accel Partners

Facebook、Etsy(工艺品线上销售)、Brightcove(在线视频平台)、ModelN(营收管理服务)、Legendary Pictures(电影制片公司)

## 02 Michael Moritz

Sequoia Capital

Pure Digital、Zappos(网上鞋类和服装商城)、A123 Systems(电池生产商)

## 03 Reid Hoffman

Greylock Partners

Groupon(团购网站)、Shopkick(移动商务应用)、Airbnb(在线零售网站)

## 04 Peter Fenton

Benchmark Capital

Wily Technology、Coremetrics、JBoss、Reactivity、Xensource、Zimbra

## 05 Scott Sandell

New Enterprise Associates

Playdom(社交游戏开发商)、Data Domain(存储服务)

## 06 Kevin Efrusy

Accel Partners

Terracotta、Aptana、Northscale(数据存储公司)、BranchOut(社交应用开发商)

## 07 Peter Thiel

Founders Fund

Yammer(企业版Twitter)、Palantir(软件公司)

## 08 Peter Barris

New Enterprise Associates

Groupon、Boingo Wireless

## 09 David Sze

Greylock Partners

Revision3(在线视频服务)、Oodle(在线分类信息)、VUDU(在线视频服务)、SoftCoin(互动式促销商店)、New Edge Networks

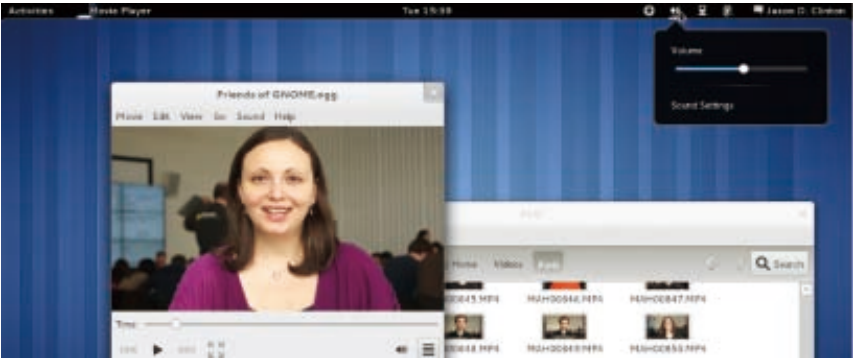
## 10 Marc Andreessen

Andreessen Horowitz

Groupon、Skype、Zynga、Twitter、LinkedIn

## GNOME 3.0正式发布

作为GNOME一个重要的里程碑，GNOME 3.0正式发布了，它适用于各类设备，采用新一代桌面用户界面GNOMEShell，并改进了GNOME开发平台（显示后端、API、搜索功能、用户消息、系统设置等）的诸多方面特性。



## 开源HTML5网页编辑工具Maqetta

IBM推出了基于浏览器的开源HTML5网页编辑工具Maqetta，支持Chrome、Firefox和Safari，提供了WYSIWYG的编辑环境。

Maqetta发布了Preview 1版，可在浏览器中直接使用，无需任何插件，用户也可以选择下载安装到本地服务器。

## 游戏优化大师

盛大旗下游戏优化大师于正式发布测试版本，这是一款为优化游戏体验而开发的通用游戏优化工具。该软件能够动态管理CPU、内存、网络，帮助玩家解决游戏中遇到的卡机、停顿、内存不足等问题。

## Amazon推出云音乐服务

亚马逊正式推出云音乐服务Cloud Player，从而也跨入了网络音乐服务的业务领域。用户可通过Amazon购买音乐并储存在Amazon服务器上，通过电脑或Android智能手机播放。目前Cloud Drive提供5GB的免费空间。只要从Amazon购买一张专辑，Cloud Drive空间便将自动升级到20GB，同时还可以储存音乐、照片、录像视频及其他数码文档。



## Entity Framework 4.1 正式版发布

微软发布了ADO.NET Entity Framework 4.1正式版。主要特性包括DbContext API，基于以前版本中的ObjectContext和其他类型抽象出的API，优化了常用开发场景和编程模式，可被用于Database First、Model First、Code First三种开发模式。Code First是基于Entity Framework的新的开发模式，用C#/VB.NET的类定义所需的领域模型，并用这些类映射到现有的数据库或者产生新的数据库结构，同样支持通过Data Annotations或fluent API进行定制化配置。

## Nginx 1.0.0 发布

现在全球排名前100万的网站有6.8%在使用Nginx，在俄罗斯46.9%的网站在使用Nginx。距第一版发布九年之久，Nginx 1.0.0 稳定版正式发布了。

Nginx是款轻量级Web服务器，同时也是反向代理服务器及电子邮件代理服务器，由俄罗斯程序设计师Igor Sysoev开发，特点是占内存少，高并发表现也相当优秀。

## Mac下的SQLite客户端 MesaSQLite

MesaSQLite是款简单好用的SQLite客户端，全GUI操作，支持大多数SQLite特性、表格、视图、触发器等功能。界面简单，管理SQLite数据文件时更方便，支持SQL直接执行。



## 本地生活信息服务应用：百度身边

“百度身边”是基于地理位置的本地生活类产品，作为本地生活信息搜索和分享平台，为用户提供优惠打折信息及消费决策支持。

“百度身边”借助了百度搜索技术，与搜索引擎结合得更加紧密，并且整合了百度旗下的地图、无线等资源。

## 系统日志存储系统Graylog2

使用了MongoDB技术的开源系统日志系统Graylog2，由下面几部分组成：一个由Java写的服务器，接受TCP、UDP及AMQP的系统日志信息；用来存储日志内容的MongoDB服务器；Web可视化的管理日志界面。



## 开源搜索服务器Solr 3.1发布

Solr 3.1是Solr 1.4.1发布之后的第一个版本，为了与Lucene 版本号同步，在版本号上做了跳跃。Apache Solr是性能强大的基于Lucene的企业级全文搜索服务器，拥有支持XML/HTTP、JSON APIs、Web管理界面等诸多功能，并运行在Java servlet容器（比如Tomcat）中。

## Red Hat秘密“Java杀手”项目Ceylon

Seam创始人Gavin King介绍了一种设计替代Java的新语言和SDK：Ceylon。该语言支持功能强大的类型系统，并考虑了以下限制：所有类型在类型系统内部都必须是可表达的（没有特殊的primitive类型或者复合类型）；通用类型系统比Java更简单；相比于其他类Java语言，安全性更高，甚至元编程也必须是类型安全的。Ceylon必须在JVM上执行，并可与Java交互操作。

## Maven中央仓库搜索引擎

Sonatype最近发布了官方Maven中央仓库搜索引擎，地址为<http://search.maven.org/>，该搜索引擎索引了整个Maven中央仓库的数据，为用户提供了包括关键字搜索、Maven坐标搜索、类名搜索、Sha1校验和搜索，以及仓库浏览等常用功能。这个官方搜索引擎的最大优势在于拥有中央仓库最新最完整的数据，索引每日更新。在搜索依赖时，用户也能看到依赖的最新发布日期，从而判断这个项目的活动状态及质量。

## 云服务平台Cloud Foundry

VMware推出Cloud Foundry，通过提供三个维度的开放架构（开发框架的选择性、应用架构服务的选择性、部署应用的云的选择性），突破了当今PaaS方案的限制，具有非标准的框架下部署，支持多种应用服务的能力，特别是可以跨越私有云和公有云进行应用部署。



## Seam 3.0正式发布

Seam开发团队宣布 Seam 3.0版本发布，并称其为Seam有史以来极其卓越的里程碑。

Seam是一系列Java EE 6可移植扩展和工具集合，通过整合一系列某一特定领域的功能模块，开发者能够创建丰富的标准互联网应用。这种模块化设计还允许用户基于项目需求，选择相应的模块，不必全部采用大型的整体框架。

# Ceylon将赶超 Java?

在2011年4月举办的QCon北京大会上，来自Red Hat的Gavin King首次公开了关于Ceylon语言的重量级话题。作为Hibernate的创始人，Gavin广为Java开发者熟知。本次技术大会上透露的Ceylon语言，是Gavin在Red Hat近两年一直投身设计并开发的新一代JVM语言，计划在企业开发领域赶超Java的领导地位。

Ceylon面向企业级应用开发设计，在JVM运行时基础上执行，支持与Java的交互操作。Ceylon语言充分借鉴了Java语言的优势并屏蔽其在设计上的不足，使用静态类型并提供高阶函数支持，在保持代码良好可读性的同时进一步降低开发者入门的难度。

Ceylon语言没有设计特殊的Primitive类型或复合类型，所有类型在类型系统内部都是可表达的。Ceylon的通用类型设计比Java的实现更为简便，而且相比其他类Java语言具备更高的安全性，Ceylon的元编程（Reflection）也必须是类型安全的。此外，Ceylon语言减少了开发者对于配置文件的依赖，屏蔽了Java开发者常常需要对各种XML进行的配置和修改操作。

Java语言的确具备众多适宜于企业开发的优点，包括良好的开放性、适合企业级开发的健壮性等特点。但是其生产效率和过重的负荷却一直令人诟病。作为一门深受Java语言影响的JVM语言，Ceylon由熟知Java语言的专家设计。正是因为Gavin多年的Java开发经验，促使他着手实现这样一个期望能够超越Java的语言。

当然，从头开始创建一门编程语言以及SDK工具包无疑是一项耗费巨大工作量的任务，但Gavin在他的Blog（<http://in.relation.to/>）上撰文写道：我们在设计开发Ceylon语言的过程中，借鉴了大量Java开源社区中的成果，如OpenJDK, JBoss

以及 Eclipse社区，并且这些工作对于Red Hat这样规模并汇聚众多开发好手的公司来说并非是不可能的任务。

社区支持对于Ceylon语言的发展是必不可少的，然而，要想获得社区的认可与追随，Ceylon语言本身能够提供一个设计优良的杀手级开发框架将是个巨大的激励因素，就像Ruby语言的Rails框架或Scala语言的Lift框架那样，在众多项目中得到成功应用。

此外，IDE环境对于开发语言的普及具有很大的推动作用，如被不少Java开发者所推崇的Scala语言，则同时拥有基于Eclipse的IDE工具、基于IntelliJIDEA的开发环境，以及NetBeans的Scala开发插件。目前，Ceylon语言也同样需要一个令人满意的IDE工具，来增强语言的适用性。

作为Hibernate的创始人，EJB3.0专家委员会成员，JBoss Seam的开发者，Gavin King已经取得了令开发者瞩目的成绩，所以运行在JVM之上的新一代编程语言Ceylon能否真正赶超Java，造就另一个辉煌，非常值得期待。P



高昂

中国标准化研究院助理研究员，从事信息技术标准化研究工作。关注开源社区，也是OSGeo中国和InfoQ中文站成员。

## 从 MIX看微软未来战略

MIX上亮点颇多，因为微软的每一新产品/技术的发布都有相当强的针对性，并且对手都十分强劲。从中可以看出微软对未来的谋篇布局。

首先来看WP7，支持远东语言输入、IE9、HTML5、Socket、多任务、陀螺仪、Camera数据实时获取，整合了XNA与SilverLight等一系列让人眼花缭乱的功能，令开发者们觉得自己的春天也来了。仔细分析这些功能会发现，微软不再像以往那样，由着自己的性子来决定下一个版本中所包

含的新功能了。这些功能背后都有实际的应用场景支持：远东语言支持，没有相应的输入法，将WP7挡在了很多市场以外；IE9发布之后，除了不支持XP外，获得了一边倒的好评，升级并支持HTML5，能够让WP7站在与iPhone相同的起点上；多任务支持，iPhone花了两年多的时间，WP7则是十

个月；Camera数据实时获取，一大票QR、条码识别的应用可以在WP7上畅行无阻了，也许增强现实类的软件也会来凑凑热闹；Socket支持，后边站着一个笑嘻嘻的KiK。

可以说，Mango标志着微软新的移动战略布局初露峥嵘。当然，微软的短板也非常明显，屡次的升级失败，难以提交应用的Marketplace，说明微软从OEM导向逐步转向Service导向，而运营经验不足，会成为微软在移动领域的胜负手。别忘了，在运营App Store之前，Apple已经对iTunes驾轻就熟了。


MIX上另一个有趣的东西就是Kinect SDK。Kinect一开始只是XBox上的体感套件，可是由于太过成功，被开发者社区“破解”后，发明了很多新玩法，比如，廉价的动作捕捉系统，而且也发布了针对PC的USB驱动。于是微软“从谏如流”，发布了针对Windows的Kinect SDK。微软长于做平台，却拙于设计商业模式，让合作伙伴去想玩法，微软来赚平台的钱。那么玩法有哪些呢？说白了，Kinect能够让动作捕捉技术进入很多受成本限制的领域。

比如，展览展示领域，Kinect可以增加展览的交互性，将

单纯展览图片、视频，变成交互游戏。目前已经有一个虚拟试衣橱软件，应用到了Kinect技术，允许用户看到自己虚拟着装后的效果。

除此之外，Kinect还有可能改变交互方式，比如取代电视遥控器，我们只要向Kinect挥挥手，就可以切换电视信号。另外在一些监控领域，我们也可以使用Kinect来监控某些特定动作。

当然，Kinect的限制也是很明显的，比如对于环境光的变化敏感、最多支持两个人、有严格的区域限制。我们在做实际开发时，还要考虑到这些因素的限制。

Kinect SDK还有很多玩法等着我们去发明，游戏开发者准备好了吗？



马宁

微软最有价值专家，Windows Mobile开发者。

不知从何时起，各大移动平台的系统代号都纷纷改成了食物的名字，苹果就不用说，Android从Cupcake开始到现在的Honeycomb，而微软也开始加入这一潮流，把Windows Phone 7的新版本代号叫做芒果（Mango）。

在4月中举行的MIX11上，微软正式宣布，芒果要来了。让我们看看芒果给WP7带来的新变化吧。

首先是硬件部分，新系统最低硬件要求中CPU从1GHz的高通SnapDragon降低到了800MHz的MSM7x30。这个变化，一方面表明了WP7在系统总体架构上的更加优化和成熟，能够在更低要求的硬件上良好运行，另一方面凸显出微软准备把WP7打入中端智能手机的计划。

这无疑对于WP7的推广是非常有利的，更加低廉的价格，能够让目标市场容量迅速扩大。这一变化对于刚刚成为WP7新成员的Nokia来说至关重要，甚至我认为，这个也是促使Nokia加入微软阵营的重要决定因素之一。而新加入的陀螺仪也越来越成为智能平台的标配，能够让应用和游戏的开发者开发出更加出色的体感应用。

对于软件部分，变化就非常大了。首先是多任务的增强（只是增强），而不是完全多任务，不过对于绝大部分应用来说，应该基本是足够了，让我们好好来看看：Background Agents负责后台加载等应用，Background Audio进行后台播放

音乐，Background Transfer进行后台网络数据或文件传输，Fast App Resume可以实现应用前后台的互相切换，某些部分甚至比iOS的多任务功能还要强大。在多任务方面的架构设计也很巧妙，决定后台运行的Agent和App的数量一方面取决于内存大小，还可以根据外接电源的情况自动增加，这样达到省电和更多功能的尽量平衡。

特别要注意的是，Agents和应用是会在不同的进程中，可以通过特定的接口进行互相通信，这一点和Android/iOS等智能平台不同，不过Agents和应用两者的权限和能够读取的数据是一样的。

在对设备的访问方面改善也非常大，一方面是新增加了陀螺仪的访问接口，另一方面增强了相机以及电子罗盘等多个体感传感器的访问接口，比如相机，原来只能调用拍照接口，不能实时获得摄像头数据，而新的API对此实现了完全的支持。把Kinect移植到WP7上？这个议题比较有趣，但从理论上来说是完全可行的。

WP7上的SilverLight版本支持增加到了4.0，而且还在

XNA的架构中实现了和SilverLight的整合，也就是说，在同一页面中，展现方式可以同时包括XNA和SilverLight，甚至二者的叠加融合。

其他部分的变动中，比较重要的是SQL CE的引入以及Sockets编程的支持。尤其是SQL CE的加入，解决了众多开发者，尤其是数据应用开发者头痛的WP7上没有本地数据库存储，只能通过文件方式存储数据的问题。

除此以外，芒果给WP7带来的变化还有很多，如更多语言的支持（16种，包括中文），更多可以提交应用和购买应用的国家（可提交应用的国家地区达到38个，包括中国大陆），IE9内核的引入，正式支持HTML5（演示中的测试远超Android和iOS，不过，各家都有一些类似自己比别家数据高的测试，所以不能完全代表实际使用情况），Live Tiles（极其像Android的Widget的功能），日历数据的访问接口等，共有1500个新的API随着芒果发布。这样大的变动，让外界流传的“Mango就是WP 7.5”的传言增加了很多说服力。

Mango的推出，标志着WP7平台的日渐成熟，确实有极大的潜力成为智能移动平台的三甲，与此同时，WP7的模拟器和开发环境也是对于用户非常友好，甚至在Mango中能够进

行一些硬件传感器的调试和模拟，再加上C#良好和成熟的语言架构，一定会吸引越来越多的移动开发者进入。

在MIX11中，Nokia虽然没有大张旗鼓地进场宣传，只是在Keynote中的单独一个Slide露出惊鸿一瞥，出现了Nokia的标志和一个手机的样机图片，但MIX刚刚结束后2天，俄罗斯一家网站就爆料说Nokia正在开发4款WP7的手机，最早会在今年秋季上市。

对于WP7，Nokia虽然有更改用户界面的权限，但其首席技术官Rich Green日前的表态中并不打算进行修改，理由是用用户体验要保持统一以及过度修改会导致升级困难。不过，个人觉得Nokia还是会继续增加更多的定制化内容，以满足各种用户的差异化需求。P



崔海斌

十几年的开发经验，专注于Java、Android、移动开发等领域。现就职于创新工场旗下的风灵创景（点心系统开发商）任首席架构师。

## Web上搞定任何事儿

在互联网上，技术的新陈代谢是不可避免的。各大公司为了构建快速网络，不断推出新的网络标准，“更快、更好、更强”是唯一衡量的准则。甚至自家曾经的标志产品，如果不能符合潮流的发展，最终也会被抛弃。

今年是FTP发布的40周年，作为互联网早期的几种核心协议，FTP与HTTP、IRC、EMAIL等地位等同。如今FTP因为安全性和一些功能缺陷，开始慢慢没落，连云存储这种典型的数据存储应用，云与终端间绝大部分的数据交换用的还是HTTP。

回想当年做网页，第一个必须掌握的技术就是使用FTP传输文件到服务器，搭建服务器FTP也是基本功。看如今HTTP大展宏图，只能感叹FTP真的老了，如同当年IRC的没落一样。技术的标准也会随着互联网的发展而更新，会有越来越多老旧的技术标准被放到博物馆里面。

与FTP相反，新贵SPDY就显得好很多，Google在2009年11月首次对外宣布了SPDY协议，这是Google构建快速网络中

非常关键的一环。在过去的18个月里面，SPDY逐渐被集成到Chrome的Stable分支中。

简单来说，SPDY是更先进、更有效率的HTTP协议。SPDY可以通过一个单独的TCP链接实现并行的多路复用流通信。并且支持优先级，优先传送最重要的HTML内容，而其他JavaScript、视频等不太重要的内容，优先级则会相对较低。总之，SPDY可以将页面载入时间缩短到现在的一半左右。要想现在体验SPDY，也非常容易：下载并安装Chrome，打开某个Google的站点（比如 Gmail），之后在Chrome 里面打开：

```
chrome://net-internals
```

如此，即可看到当前活动的SPDY进程。对于使用者来



说，SPDY与HTTP没有任何区别，我们也很难“看”出使用了SPDY后有什么改进，但是我们肯定可以感觉到Google服务在Chrome下异常快，这就是SPDY的功劳了。


相比之下，软件更新就频繁得多了，微软的IE最近有了大动作：IE10预览版发布。对此预览版来说，头一个让人感到奇怪的就是不支持Vista，而微软做的软件不支持自家的操作系统，却大概不是第一次了。虽然每次都感觉很神奇，甚至认为它们是不靠谱软件的代名词，但这次的不支持恐怕和微软制定了XP的“死亡时间”一样，是“别有用心”之举。

微软称，IE10与IE9一样，也建立在硬件加速图形工具软件之上，并将继续支持HTML5和CSS3。据说，网络开发员最常用的两款CSS3指令，Gradients和Flexible Box Layout都已经成为IE10的组件。

令开发人员不解的是，微软大力支持HTML5，为什么还费力推SliverLight 5呢？虽然称SliverLight 5完美支持IE9，但HTML5如此风光，想必SliverLight 5的开发团队也有些三心二意了。

看来支持IE9这种富B/S结构的应用前景会更好，就应用

的方式来说，现有HTML5、Flash、SliverLight等技术，现在平板大行其道，Flash因为纯粹靠CPU来运算，耗电量巨大，苹果对其封杀再三，而对SliverLight却懒得封杀，足可见其未来毫无威胁。对于HTML5来说，只要能确定一致的标准，一统江湖是一定的，但目前还需要时间来打磨。

如今HTML5编辑工具相对缺乏，IBM无疑做了一个巨大的贡献：它发布了开源的HTML5可视化设计工具Maqetta，这对于开发者来说将是一个福音，虽然此举会将Borland挤得无处生存。并且听闻，Maqetta源码已经贡献给开源机构Dojo基金会，既开源又免费，看来IBM已经了解互联网的游戏规则了。想想以前，HTML5与AJax的编写和调试都极其麻烦，这种集成性的工具将大大提高开发效率，Maqetta的项目经理说得好，“我们的目标就是在Web上搞定任何事儿。” 



钱宏武

现任职盛大创新院，原搜狐互动产品开发部主管，资深互联网社区架构师，垂直搜索领域专家。

## 甲骨文和“安腾”蜜月结束

近期数据库市场虽然热点不少，但前期基本上都是各类“龙套们”在热场。因为恰在“三大”数据库产品发布的中间期，所以“三大”的产品较少，而各类周边的数据库工具、NoSQL平台就成了新闻的主角。时至4月初，一则甲骨文和“安腾”分家的消息不禁让很多用户闹心起来。

甲骨文官网轻描淡写地说放弃安腾的原因——看不到关于“安腾”长期的战略方向。不过作为过渡，甲骨文还会为客户提供既有“安腾”版本软件的支持。

如果说此前微软、红帽放弃“安腾”只是令部分用户略感唏嘘的话，此次甲骨文放弃“安腾”则是对该平台一记重击，作为佐证，甲骨文发布的声明中称英特尔也在预见“安腾”的末路。

“安腾”的失败一方面因为非主流技术趋势要为主流技术惯性让路，另一方面可以说“安腾”生不逢时，2000年之后随着互联网第一次产业热炒的没落，用户开始真正看清“业务”而不是“技术”才是信息化的目标，不能本末倒置，而“安腾”却逆市而动，强迫用户升级软硬件系统。

如果说它早出生三年，也许会成为用户争先购买的“时

髦”产品，但身处互联网严冬的“安腾”，它的热度就像一根火柴一样，绚丽一下之后就渐渐黯淡了。

从甲骨文此举的效果看，损失最大的也许不是英特尔而是惠普，甲骨文最终选择了自家的SUN服务器，至于“安腾”只是双方“感情不和”的一个不大不小的借口。

也许对于一些用惯了IBM小型机、信息化“不差钱”的单位，这个消息只是茶余饭后的聊资，但对于之前大量使用HP-UNIX和甲骨文组合的用户而言，这个消息很成问题，除了更换服务器、操作系统外，还可能涉及存储、监控软件和集群工具。另外，最棘手的是自主开发的企业业务系统，虽然都是基于甲骨文数据库，但各种后台任务脚本、PL/SQL暗示在不同的操作系统平台多少还是存在差异，此外C++、Fortran和Java编译器的差异其影响更为隐蔽。

其实甲骨文对于HP和SUN的扬弃已经体现得非常明显，

第一代Oracle Exadata选择的是惠普方案，简单试水之后，甲骨文发现同时控制硬件、中间件、数据库和咨询后的甜头，随后在完成对SUN的收购之后，甲骨文第一时间将自己控制的硬件打包到Exadata V2中，此举不仅叫好更叫座，以至于Exadata V2一度出现“一机难求”的局面，这也从另一个侧面体现客户对于SUN服务器的认可。

此次以“安腾”为借口被抛弃后，惠普其实面临的不仅是甲骨文数据库一个产品的阻击，从效果看它面临的是前BEA的WebLogic、甲骨文的Oracle数据库、前SUN的硬件和存储、前SUN的Java EE，同时英特尔似乎也准备抽身而去，这些都令惠普服务器产品的情势很严峻，以至于惠普执行副总裁戴夫·多纳特利回应此事时用了“无耻”一词。

对于甲骨文而言，此役之后就是IBM了，甲骨文暂时还不具备在小型机、大型机领域打击IBM的准备，不过“云计

算”也许是绕过马其诺防线的捷径。

且不管商家的唇枪舌战，站在用户角度，尤其是国内占比很大的甲骨文用户而言，如何应对才是要点。

首先，放弃“安腾”是不得之举，加紧在目标平台的确认证测试成为当务之急；其次，应该适时联系遗留软件、系统的供应商，尽可能早的准备退路，否则真到切换档口再准备恐怕就要付出血本、前途多舛。P



王翔

软件架构师，主要研究方向为XML、.NET、领域设计和PKI应用。工作之余喜爱旅游、写作和烹饪。

## 开源硬件和开源云计算平台

关注开源技术和企业开放平台和开放API，关注淘宝、开心、人人、新浪微博和腾讯微博等开放平台和电子商务、企业管理软件结合的机会。关注开源技术，是为了节省成本和提升自己公司代码水平；而关注开放平台，是想草根创业，所有的一切，都应该站在巨人的肩膀上。

偶尔也会关注一下炙手可热的Facebook，也经常会听到很多软件公司和电子商务公司围绕Facebook做插件开发或者进行电子商务方面的营销。同时，我们也知道，Facebook也已经贡献了自己最知名的NoSQL数据库Cassandra给Apache开源社区维护。

事实上，对于开源，平常我们听得最多的也都是有关操作系统和应用软件方面的内容，但最近Facebook却史无前例地针对数据中心和服务器推出了全开源项目——The Open Compute Project，将开源“开”到了极致！

此项目是由Facebook牵头，与惠普、戴尔、AMD和英特尔共同合作，它包括数据中心和服务器两部分。惠普行业标准服务器和软件业务首席技术官Greg Huff说：“拥有庞大计算需求的企业仍将寻找创新技术，以便突破当今的障碍。”戴尔服务器平台副总裁Forrester Norrod表示，他的公司已经根据Facebook的新设计为一些企业提供了部分系统，“这意味着你将拥有一个开放的创新平台。”

Facebook的数据也表明，他们的定制硬件各方面指标比业界标准要高得多，与公司从设备厂商购买的同类现成产品

相比，效率提升38%，成本则降低了28%。而整个数据中心的能耗按PUE（Power Usage Effectiveness，电能使用效率）衡量是1.07，大大低于业界通常的1.5。

这的确是一个非常有意思的话题，一位心理学博士（扎克伯格）去做SNS网站，不仅领导开源了自己的软件（Cassandra，NoSQL数据库），居然还开源了公司的硬件架构。这说明技术的推动和革新，并不一定全部都是由专业人士完成的，第三方需求，同样能够推动软件技术和硬件技术的创新。

不过，反观国内，像中国移动、电信、大小银行，他们对数据中心的需求，肯定也不会比Facebook差，为啥移动、电信、银行就没有出现这样的软件和硬件创新呢？为啥非得购买Oracle、Sybase数据库？也许体制压死人。

如果说硬件开源很拉风，能炒作，那Vmware开源了PaaS平台Cloud Foundry，就有点波澜不惊。值得一提的是Cloud Foundry能够广泛支持包括Spring for Java、RoR、Sinatra for Ruby和Node.js等框架；也支持其他基于JVM的框架，例如Grails；还支持多种应用服务，包括MySQL、MongoDB、



Redis以及VMware自己的vFabric应用服务。

说实话，个人不喜欢万金油Spring，所以看到Cloud Foundry也似乎是个万金油，心里不免也有点犯嘀咕，感兴趣的同学，自己去尝鲜吧。P



邢波涛

现任北京新软孚信息技术有限公司技术负责人。关注SaaS管理软件和B2B、B2C电子商务的融合。

## 一个游戏 AI编程的想法

游戏中，要么和其他人交互，要么和电脑交互。在人与电脑的交互、对抗中，电脑表现出的行为和处理事务的能力，很大程度上影响游戏体验。目前对于游戏NPC AI处理的通常方法是什么呢？

通常的NPC控制，大体都是围绕着状态机来具体实现，程序员把NPC（这个NPC是广义上的NPC，一个箱子，一个机关都有可能是NPC）的行为，分割成很多可以穷举的状态。比如人类，他就可以分解成站立休息、走路、奔跑、攻击、防御和被击中等等很多状态。我们AI处理所要做的事情，就是根据不同的条件，将角色置换到对应的状态，而状态本身来处理本状态下的各种表现和事物处理。

状态机代码大体如下：

```
Class State{
    Virtual BOOL Enter(arg); //进入状态调用
    Virtual BOOL Leave(arg); //离开状态调用
    Virtual BOOL Tick(arg); //当前状态轮询维护
}
```

然后各种状态，派生自接口，实现各自的行为，比如Tick过程发现HP变成0，就把状态置换成为死亡状态等等。状态机之间能够正常翻转起来，一个NPC的行为就完整了。随着NPC行为的越来越复杂，状态机状态就越来越多，状态转换的行为也会越来越复杂，这样，要维护这个复杂的系统，往往是游戏内，除开引擎外，最复杂，最消耗精力的。这个系统牵涉到方方面面，而且结构复杂，逻辑过程一旦出错不容易调试，而它又偏偏是一个很容易出错的系统，通常解决这些问题的都必须是项目内优秀的程序员，这样成本非常高昂。我们有没有更好的方案来解决这个问题呢？

我对于这个问题的思考，在这里探讨一下。目前我们游戏内遇到的AI行为，最主要的就是，在某种条件下，AI控制单元要做对应的响应，响应通常就是改变自身，或者别人的状态。

编辑器数据和UI的绑定。有了Reflection特性，我们可以做一个统一的规范，让编辑器界面和程序内数据结构直接映射，减少大量繁杂的关联代码。

游戏对象的动态创建和初始化。在MFC一类的类库中，通常会提供一套RTTI（运行时类型信息）来做动态创建，规则复杂，而且创建和初始化参数等处理起来就力不从心了。

游戏内功能模块组合配置化。比如我要做一个流程处理，某种情况下面调用某些代码工作。但是这个调用我希望能够灵活配置，那么我们可以通过一个配置文件，或者数据库字段，获得处理函数入口和参数。这样，后面的各种功能配置组合，都可交给策划去填写配置文件或者数据库字段了。

类似游戏内图文排版一类需要高度灵活可配置的模块，用Reflection是一个很不错的选择，可以比较轻松面对各种调整修改。

另外，在.NET平台，配合Reflection，还有一个Attribute系统。这个Attribute的作用，就是可以对用户所写的代码，进行一些标注。比如我可以给一个类添加一个Attribute，给它打上一个标注，这个类是用来描述NPC行为的，然后再给这个类成员函数标志一个Attribute，申明这个函数是用来处理NPC被交互的时候调用的，等等类似。然后我们可以自己写程序，搜集程序集内，所有打上这些标记的类和函数，然后统一做逻辑处理。

那么在传统C/C++开发平台上，我们能不能实现类似功能呢？答案显然是可以的，依靠C++强大的template和macro，这个过程甚至可以做到基本全自动。如果读者对于在C++程序中实现一个强大的反射系统。我下个月可以和大家一起讨论一下相关话题。P



宋忆疆

参与《碧血情天》、《傲世三国2》、《乱舞天下》等游戏研发。目前担任《流星蝴蝶剑OL》项目制作人。

# 让天下没有难卖的软件

## ——实记CSGS2011

记者 / 付江

4月15~16日，由电子商务解决方案提供商Digital River携手CSDN共同打造的“中国软件全球营销论坛”，如约到来。超过300名知名软件开发商代表、技术高手、重量级营销专家齐聚一堂，分享了最新软件商业模式和营销技巧。

大会演讲嘉宾，既有在IT领域摸爬滚打十余年的资深专家，也有在移动互联网时代开发出畅销软件、迅速致富的新贵。此次盛会包括Digital River集团众高管对全球软件营销趋势和机遇的深入分析、CSDN&《程序员》创始人蒋涛的“中国开发者的黄金十年”、SWREG总监贺大维的“基于互联网提供软件服务的模式”等十多场精彩演讲。

### 在线营销将成主流 移动互联网带来新冲击

DR globalDirect副总裁艾马克的演讲尤其引发了记者的兴趣，他通过一组组具体数据解析了全球电子商务产品和在线购销的趋势，以及国内软件开发者的海外市场机遇。首先将两组图和数据对比起来看：



表1 不同地区软件供应商对DR收入的贡献率	
Vendor Contribution to DR Sales	
INDIA	4.29%
AUSTRALIA	3.27%
CANADA	4.17%
RUSSIA	8.73%
ALLELSE	7.08%
CHINA	15.38%
EUROPE	37.48%
Total	100.00%

如图1和表1所示：美国消费者在线（付费）购买占到47.35%，但是来自于美国的软件供应商对（DR的）收入贡献只占到37%；欧洲用户占到28.3%，但是他们对收入的贡献只是19%；中国的用户只占到1%以下，仅仅为0.45%，但是中国供应商对收入的贡献却占到DR的15.38%。这意味着，广大海外市场消费者在线购买的软件大量来自中国供应商。换言之，如果中国开发商没有住在美国或欧洲，未来成功的最大机遇就是向美国和欧洲地区销售自己的软件。至少短期来看，那边的用户付费购买软件的习惯和大环境更为良好。

根据统计，美国市场2010年在线销售额比2009年上升了15%，达1790亿美元，并且在未来仍将保持上升趋势。在消费者选择购买的软件类型上：系统相关的占大头，个人消费品如音乐、电影和数字杂志等软件产品和休闲类游戏都呈明显上升趋势。

随后，Digital River 销售经理溥可瑞围绕移动互联网时代软件营销新策略，介绍了DR的移动策略：App Store已经成为软件分销的重要渠道，DR帮助客户建立了Android和WP7的应用（渠道平台），让客户通过系统来进行应用的销售，同时DR有能力实现这种微付费。此外，DR通过联署的网络营销，为客户提供更多的选择，让他们把应用更好地推向市场。







CSGS 2011营销论坛上，各种专家济济一堂

## 国内软件海外营销技巧

Digital River GmbH部门经理唐雅娜分享了（在平台上销售软件）“优化转换率的最佳实践”心得，她的建议包括：到达页面的导航要非常简单；在沟通方面要非常清晰，所有的信息都要简明扼要，最好是用事实、例子、视频甚至是一些小样；还可以使用如对比、图表、检查清单等工具；给不同的应用以不同的信息；描述产品的时候要非常清楚，所有的版本都列出来；提供个性化的价格和营销策略；要对店铺的优化效果做测试；收集用户的联络信息和反馈；搜索引擎的营销等。

对于Google“咖啡因”新算法架构下的成功策略，如何能更好地被搜索引擎捕捉，DR globalDirect渠道行销经理蒲丹妮认为，重要的是保证网站有独特的内容，并持续更新，不要做成“内容农场”。此外，她还建议在网站中加入视频元素。“尤为重要，不要让网站里全都是广告。”

苏州思杰马克丁软件创始人金勇在“货卖一张皮”的演讲中，重点强调了UI的重要性和设计技巧。他认为海外市场的UI设计分为两个方向：以美国为代表的北美系和以德国为代表的西欧系，“这两个市场目前占到自己所做过产品的八成到九成。”从目前趋势来看，以简洁易用为代表的北美系逐渐成为主流，甚至加入了工业设计。“用户生活节奏越来越快，很多人通过Google、IM、Facebook，或朋友介绍，寻找某个问题解决的小工具，大量用户其实并不需要太复杂的功能。”

## 国内软件开发商分享创业心得


无锡阿达游戏软件公司董事长顾方除了在演讲中探讨了

如何打造一款成功的互联网软件，还分享了自己的创业心得和对成功的看法，他认为“成功是一个过程，如果在追求成功的过程中不能快乐，最终这个成功所谓的点就不会到来。能够选择一个让自己快乐的题材去追寻成功，追寻自己的理想，这才是我想要的成功。”

其他嘉宾的演讲包括，谷歌中国商务解决方案部出口行业负责人张山峰的“海外网络营销新趋势”、Softonic全球销售总监斯蒂文·克兰斯顿的“有效运用全球广告机会的成功秘诀”、华军软件园主编米晓斌的“IT业迎来软件时代”等，都获得了与会者的频频响应。

在论坛环节，米晓彬、傅宇权、顾方、金勇、张山峰几位嘉宾就创业心得、创业中遇到的困难、如何和投资者打交道、对未来软件营销模式以及对于成功的理解等话题展开了充分探讨。

## 编后语

Digital River产品管理副总裁李浩明演讲中的两句话可以用作本文的结尾：“首先，技术上最关键的是必须站在客户角度考虑，即所谓的客户至上；第二点就是要考虑产品的价值。作为一个技术专家，职业生涯的早期，我总想要用那些特别酷的技术，最后发现最重要的就是钱，怎么样能够拿到钱呢？必须要增加价值，无论做什么业务，都必须把客户放到中心考虑，并且要为客户增加他们的价值，如果你不这样做的话，你的工作就没有做到位。”

# 春天里的平板电脑和数据中心

## ——2011英特尔信息技术峰会有感

记者 / 董世晓

伴着暖暖的春风，2011英特尔信息技术峰会（IDF）于4月12日~13日在北京举行。本届IDF以“智无界，芯跨越”为主题，展示了英特尔从硬件、平台到软件和服务的全面的计算解决方案，推进了个性化互联网的发展，同时也将面向中国市场支持本地化创新，助力新一代信息技术等战略性新兴产业发展。

### 重新定义平板电脑

毫无疑问，移动互联网的黄金时代已经到来。而要想享受这个黄金时代的乐趣，移动设备不可或缺。因此在本届IDF上，从英特尔高层到普通参会者热议的话题无疑是平板电脑，以及英特尔力推的MeeGo操作系统。

尽管此前遭遇了与诺基亚在MeeGo系统上合作的决裂，但在本届IDF上英特尔向外界传递了一个清晰的信号——与诺基亚的合作，仅仅是MeeGo战略的很小一部分，英特尔将抱着更大的决心和意志与产业链各方进行更加紧密的合作。

在平板市场中，iPad凭借绝佳的用户体验，得到了“果粉”的追捧，而Android平板则高举开源和性价比大旗，为自己拼得了一片天地，可以说一时间硝烟四起，竞争惨烈。作为有志于在平板市场大展拳脚的英特尔，它的发力却总给人一种不瘟不火的感觉。对于这种情况，英特尔此次旗帜鲜明地表明了自己的态度——要做就做到最好——正像英特尔副总裁兼平板和上网本设备事业部总经理Doug Davis指出的那样：“如今市场上的平板电脑都不是完美的产品，英特尔将重新定义平板电脑。”

平板电脑作为用户的“贴身小棉袄”，必须提供完美的用户体验。英特尔高级副总裁、软件与服务事业部总经理Renee J. James从“用户体验决定决策”的思路出发，指出提高用户体验需要实现多种设备之间的一致性，打造一个互联计算的世界，具体可以从芯片、平台、操作系统、应用与服务四个层面进行改善。而通过英特尔AppUp中心、与腾讯建立联合创新实验室等形式，可以使英特尔更专注于开发绝佳的移动互联网体验。

平板的竞争才刚刚开始，而英特尔的加入，为这个本就不平静的市场带来了巨大的波澜。不过，竞争的结果肯定会惠



2011 IDF如期而至

及广大用户，更多用户体验好的平板电脑将进入寻常百姓家。

### 云计算数据中心的参考架构

随着互联网数据的爆炸性增长，有权威分析机构预测，到2015年，全球将有超过10亿网民、超过150亿台终端设备接入互联网，这无疑为面向云计算的数据中心提出了更高的要求，也带来了如何保证数据的安全性、可管理性、可扩展性以及如何确保未来数据中心建设的开放性和可选择性等一系列挑战。正因如此，英特尔日前宣布了“2015云愿景”，而在本届IDF上英特尔更是展示了8款经过验证的云计算数据中心参考架构。

据英特尔数据中心事业部高密度计算业务总经理Jason Waxman介绍，英特尔目前已有超过30套通过云构建计划参考架构交付的解决方案，涵盖华为、微软等众多IT领袖。这些解决方案所展示的内容既包括对于云计算部署的配置，也包括具体到数据中心所面临的挑战，如：能效管理和用户端优化等。

### 后记

移动互联网和云计算作为当下最流行的IT趋势，已牢牢地吸引了公众的眼球。而作为这个大趋势的组成构件——平板电脑和数据中心，它们的发展像移动互联网和云计算一样，正沉浸在生机勃勃的春天里。P



# 移动资讯TOP10

## 1亿移动设备俱乐部：OS竞技场中谁是赢家？



Vision Mobile报告，智能手机在去年表现惊人，占有了超过20%的市场份额，但是主流市场还是被非智能手机牢牢占据。2010年，Android的出货量超过了苹果的iOS和RIM的黑莓，直逼诺基亚的塞班系统；OEM厂商中，索尼爱立信和摩托罗拉败局已定；LG增势缓慢；三星轻松保持着市场第二的佳绩。诺基亚仍然占据移动市场的霸主地位，去年有34%的市场份额，RIM和苹果公司各占16%。

## 连线：移动互联网最大商机不在手机程序

大商机不在智能手机程序，它只是存在于简单的“连接”中，数十亿人只是想“联系”。只有少数产业能服务我们，并从我们手中赚到钱，赚钱不会从智能手机程序开始。iPhone只占全球2%的市场，与短信和多媒体信息服务相比，它的营收只是九牛一毛。去年，全球共发送6.1万亿条短信，每秒

20万条。全球共有42亿SMS用户，想象一下：只要其中有一百万或者两百万人付你几便士的情形。通过SMS，你可以传送许多东西，现金、机票均可。最赚钱的也不会是地理定位服务，或者是视频，Ahonen认为数十亿人只是想“联系”，不论是SMS服务或者是旧的WAP服务，只要能“联系”就行。如果你提供此类服务，人家就会买单。

## 专访FIT创始人：创业最大挑战是内心浮躁

FIT输入法是一款在iOS平台上很受欢迎的免费中文输入法。从2006年的Fun Input Toy，2008年的WeFIT，一路发展至今，FIT已成为覆盖Mac、iPhone/iPad、iPod Touch端的全系列产品，FIT背后的研发运营也从最早只是创始人的兴趣玩物成长到专门成立公司运营。最近他们的一个重要动作是延伸应用FIT写字板在App Store上架，新点科技还处在初级的创业阶段，但由于员工都是从2008年做外包业务时就开始磨合，所以非常有凝聚力和战斗力。

## 2011年：签到已死？

作为移动互联网最受关注的热点之一，各式LBS应用一度大量涌现，但其发展局限也越来越被更多的业界同行清楚认知。LBS只是一个功能特性还是可以支撑起一个产品？签到如何添加黏性和用户核心需求结合？美国创业企业Goby联合创始人兼CEO Mark Watkins在ReadWrite上发表文章，深入探讨了

LBS的发展形式，认为“签到”只有找到更深入的用户需求，使更广泛的大众接受，才能改变消亡的命运。

## Camera+销量突破200万 创始人分享成功经验



创始人John Casasanta指出，他们花了6个多月实现销量突破100万大关，而接下来3个月就销售了200万份，照这个趋势下去，300万销售大关也许只需要1个半月。当然这是最乐观的估计，Casasant同时指出Camera+的In-App purchase收益也很不错，比如一个模拟拍摄效果的插件，卖出了超过9.8万份，仅仅4个半月就带来了7万美元的收益。Casasant解释了tap tap tap决定一款产品是收费还是免费的方法：原则上，如果绝大多数人能够利用好，就实行免费；而有特殊需求的，就可以考虑用In-App支付。

## [评论]Google的困境：永远beta式研发不灵了

美国沃顿商学院Knowledge@

Wharton网站于3月30日刊文“Is Google Stuck in 'Perpetual Beta'?”，分析了Google目前面临的问题，认为曾经在互联网上助其攻城拔寨的“永远beta版”研发模式在智能手机和平板电脑等硬件领域未必能够奏效，目前Android的麻烦很多也来自这种研发文化。

## HTC市值达到339亿美元 超过诺基亚与RIM

4月7日，台湾手机制造商HTC称，HTC市值首度超过了手机巨头诺基亚，这对HTC而言将是一个重要的里程碑。在此之前不久，HTC的市值刚刚超过黑莓制造商RIM。目前HTC的市值已经达到338.8亿美元，高于诺基亚的328.4亿美元，也远远超过了RIM的285亿美元。业内人士对此均表示吃惊，因为目前HTC的市值是5年前的30倍以上。高盛报告显示该公司的市值在未来还有望进一步增长。

## [讨论]平板首战：iPad胜出 Android完败！



虽然苹果没有透露过具体的iPad出货量，但有数据显示iPad 2在3月份出货量在250万台左右；2010年iPad仅在5天内就卖出45万台。与此相反Android平板销售惨淡，其中最被看好的Xoom，德意志银行估计目前仅售出10万台；三星也被迫降价应对。Apple以稳定的更新步调按时发布了iPad 2，软硬件上的更新虽然算不上革命，但在用户体验上的改进有诸多亮点，同时Apple的低价极大地压缩了空间，又以巨资最大程度地囤积关键零部件，更使竞争对手痛苦不堪。

## 腾讯能否挽救被诺基亚抛弃的MeeGo手机操作系统？

英特尔与腾讯公司最近宣布在上海联合推出一个合资的MeeGo创新平台，运行英特尔Oak Trail处理器。双方在2011

年IDF北京大会上宣布此事，虽然这项声明当时并没有引起人们的注意，但该合作背后的原因却颇耐人寻味。双方希望通过此举为MeeGo移动设备引进应用程序，而且将首先引进游戏内容。游戏中间件制造厂商Razer和腾讯将把四款知名的在线游戏及6亿多每月用户引向手机平台，MobileBeat网站的Nicole Scott认为，虽然此举并不能有效重振MeeGo，但至少说明这个手机平台的生命仍未终止。

## 消息称RIM欲和Google争夺北电专利

加拿大电信旗下的北电公司拥有6千多项专利，其中包括无线服务、数据网络和LTE，专利购得者将能够控制和授权黑莓、苹果和Android操作系统所需的技术。4月初Google欲以假马竞标的方式出价9美元亿买下了这些专利，美国时间4月16日Bloomberg称，两名不愿透露身份、熟悉RIM竞标计划的内部人士称，RIM正着手准备在北电无线技术专利上打败Google，同时有一些包括手机制造商在内的企业，也在筹划联合资金阻止Google竞标成功。如果有第二家企业参与竞标，那么需至少出价9.29亿美元。

## 移动照片应用程序Color：重塑社交互动？



初创公司Color在正式推出产品前就从红杉资本、贝恩资本、硅谷银行处筹集到了4100万美元资金，这是红杉资本迄今为止最大的一笔“推出前”投资。

Color使用了一些非常有趣的技术，对任何距离手机100英尺范围内的人来说，本机上的所有照片都是公开的。而且即使在没有手机信号、接收不到GPS的环境中仍能正常工作。业界人士认为，Color的未来虽然还不明朗，但是很有希望。有了这笔新的资金和27名工作人员，Color现在有足够的时日来磨砺他们的产品。P



# 苹果App Store引发的悲喜狂欢

文 / 郑金条

日前，关于苹果更改应用排名算法的推测和限制App Store内部广告代理商奖励式下载的行为，在传媒界和开发者中引起了轩然大波。

在应用曝光方面具有专长的Fiksu称根据他们的数据监测，苹果4月14日午夜之后就对排名算法进行了调整，这个调整据说更改了以前更多依赖下载基数为排名依据的算法（尽管这个算法本身也并不透明，但是整体上还是以近期下载量作为加乘权重进行计算）。监测方Flurry是以Facebook应用排名的蹿升（美国榜）、W3i以Angry Birds Rio排名的暴跌（德国榜）等各种典型因素来推测苹果App Store排名算法的调整，其关注重心可能从原来的下载量转向用户黏性（可能是根据用户在所下载的应用上逗留的时间或者他们对这些应用的评级进行测算）。

## 源于免费模式大行其道

奖励式下载模式在苹果应用商店这么火爆的关键源头还在于免费模式的大行其道。可能连苹果本身也没有想到App Store付费方式的转接这般迅速，根据Urban Airship测算，目前支持虚拟交易的应用已经高达31%，而今年GDC会议上也有一个共识认为免费模式将成为手机App中的重要营收承载。

知名分析公司Parks Associate早先的数据也称虚拟交易和广告展示将成为手机平台的主流趋势。Read Write Web在对2011年的手机应用市场进行预测时，也将免费模式中的虚拟交易当成最值得关注的选项，并称该模式在2011年将具有非凡的发展潜力。来自Bango的数据也



有力佐证了这一点，虚拟交易将在手机应用营收中占据三分之一的比例，持更超前观点的是Juniper Research，它认为2013年手机市场虚拟交易产生的营收将超过付费下载。

## 奖励式下载的潜在市场

奖励式下载广告模式很重要的一点是依托在免费模式的趋势下获得了不俗的发展空间。大量的游戏虚拟货币在免费模式催生的虚拟交易中获得了价值。在过去的半年中，Tapjoy、MdotM、G6 Pay、Adknowledge和Flurry等代理公司在奖励式下载模式下协助大量的应用获得无数的下载用户。这种激励机制在契合用户不付费获得虚拟货币的心理需求下很快获得了病毒式的蔓延。相当多的免费模式游戏都是奖励式下载广告范畴的受益者，比如Glu Mobile的Gun Bros（Michael Breslin曾赞誉过Tapjoy在游戏下载营销中的重要作用）。

但是很显然，苹果对奖励式下载的限制，可能是这种非自然的奖励下载行为被认为有可能操作应用的最终排名，并有可能使一些不是很优质的应用出现在靠前的位置而损害苹果商店的公正商誉以及用户的下载体验。

Flurry副总裁Peter Farago称此举将在苹果和用户之间建立互信桥梁（开发者或者平台失去用户和评论家的信誉才是更为致命的），尽管将影响现有利益获得者（广告行销代理执行商、优势资源型的开发商和发行商），但对于整肃App Store目前所形成的资源/资本型环境将具有相当的改造力度，使那些好创意和高质量的应用有可能在不需要耗费巨额营销资金的基础上靠苹果的推荐渠道获得更好的曝光。游戏邦认为苹果在应用程序级数达到40万量级的时候需要对整个生态圈做更加完善的改进才能维持持久的竞争力，因为寡头市场和资源市场将伤害更多独立开发者介入App Store的积极性。

当然反对的声音也必然不少，因为很明显这些广告商所做的确实绕过了苹果的营收范畴，也明显限制了苹果自身的广告体系IAD的表现。《僵尸农场》开发商The Playforge就认为此举并不能解决iOS生态圈中的应用曝光率问

题，事实上也是这样的，即使苹果做出了看起来公正的举措，但是你会发现目前整个排行榜的格局并没有多大的变化，Angry Birds不可能掉出前十，而你刚刚推出的应用也很难出现在排行榜内。

## 关于应用黏性的说法

本次调整关注的核心就是应用的黏性。Flurry副总裁Peter Farago认为在考察用户的黏着度（访问频度和时长）和下载量方面，开发者似乎更愿意选择直观的下载量（看起来下载数据更容易打动人）。但是下载量其实会掩盖应用或者项目运营的非健康状态，因为合适的病毒式渠道或者名人效应传播都可能促成一定范围的下载，但这些保证不了产品的后续影响。

---

苹果调整App Store算法并限制非自然下载有利于行业的改进，对于独立开发者而言是一个好机会，但是独立开发者依然要通过各种营销手段，才有实现产品成功的可能。

---

一个很惊悚的数据，来自Localytic的研究认为26%的下载应用会成为用户的一次性尝用品；更惊悚的是Flurry调查总结出的行业标准则，即下载1个月后，用户留存率为19%，而3个月后，用户留存率则不到5%。

因此各方面都在呼吁手机应用市场的评估标准要从单纯的下载量迷恋中脱身出来，用户黏着度才是关键中的关键。事实上有一些手机应用公司的活跃用户都让人瞠目结舌，Blackflip Studios月活跃用户为2500万、Newtoy（Zynga）旗下Words With Friends月活跃用户超过500万、Angry Birds月活跃用户为4200万和Capcom Mobile月活跃用户为640万。

## 关于黏着度的产生

什么样的游戏才有可能有理想的黏着度？

早先有段时间，Andreas Illiger旗下的Tiny



Wings几乎横扫了苹果App Store在全球各个国家和地区的榜单。这款只需要一个小指头控制小鸟飞翔翅膀的游戏，在梦幻画风的衬托下俘获了无数用户的心，简单和闲趣是很关键的一点，游戏邦觉得更重要的是它挖掘了每一个玩家心中不泯的一个小小的飞翔梦想（You have always dreamed of flying）。并且0.99美元的设置也刚好在玩家冲动消费的范畴内（Lower-priced和High-concept）。

当然，这个只是偶然才有的案例，类似Tiny Wings的游戏长时间主宰手机游戏世界的情况并不常见。

但做出纯粹的游戏是很多开发者共同的心愿。首要的一点是把游戏变成为进程观察娱乐而不是变成繁琐的操作在很多成功游戏里都获

得论证，比如Tiny Wings或者Angry Birds。在这一点上，我们可以从三家超级公司里获得一些言论支持：Rovio公司Ville Heijari认为简单永远是第一要素；PopCap公司Andrew Stein认为需要在第一时间让玩家融入游戏；Chillingo公司Joe Wee认为玩家的感情投入对于游戏的成功至关重要。

Bolt Creative工作室Dave Castelnuovo有更进一步的说辞：

■ 手机游戏是个重视轰动效应的领域，开发者如果没有办法发现和制作这个引爆点将很难在最大范围内推动这款游戏的曝光度；

■ 如果你的游戏能和你的内心产生共鸣，那么它也能与用户形成交集。

而对于非单机游戏而言，用户交互功能和信息分享系统是促成社区化黏性的关键因素，你需要依托整个社交网络架构将用户的交互需求串联起来，分享的乐趣将是促动玩家继续游戏的外部推力。

当然，没有一劳永逸的事情，如果开发者想取悦一个平台的玩家，最好的方式是针对某个平台量身打造，简单粗暴的移植很难奢望能够出现量级的下载，除非开发者或者游戏本身已经是超级品牌，有无数的外部力量自发推动你的曝光。

此外，开发者可能还需要关注的是玩家没有对任何游戏忠诚的义务，市场的可选择性将让玩家异常挑剔。因此除了简单闲趣和社区化黏性方面的努力，应用的稳定性是另外一个前提，没有稳定性的黏性事实上并不存在，而用户信息安全性、画面风格、应用的启动速度及少占用玩家本地资源都是不可或缺环节。

Neil Young曾经对外称关注用户体验要像做电影一样做游戏，因此最后的一点是你还需要人性化的用户服务，除了让用户不挑剔，用户的意见可能是修订你应用最佳的建议。

关于App营销

上文提到苹果App Store排名算法的调整有可能将创建一个除了资源优势和资本优势之外相对公正的平台环境，有可能使那些拥有好的



创意但可能没有规模营销资金的应用获得了新生的机会。

但就如我们上面的分析，Angry Birds或Cut The Rope这样的好游戏并不会掉出排行榜，而一个独立开发者推出的应用也很难像Tiny Wings那样一飞冲天。这种调整呈现的效能可能需要一段时间的观察才能做出更好的评估。

这同样意味着，开发者推出的应用在这些超级商店里还需要为了一个推荐的位置争得头破血流。在今年的SXSW大会上Patrick Mork解析了一个数据：苹果App Store应用超过30万、Android Market应用超过20万、黑莓App World应用超过1.2万、微软WP7应用超过1万以及惠普Plam应用超过0.4。

对于拥有无数应用的商店而言，任何改变对于商店的整体环境可能是有益的，但是分摊到个体开发者身上你依然会发现这样的改变其实跟你关系不大，如果一定要说有关联的话，那一定是：改变前，独立开发者可能完全没有机会出头，改变后终于出现了出头的可能性（尽管这个可能性是多少很难评判）。

所以，作为开发者，你仍然绕不过营销。

■ 开发者需要一个合适的发布时间，在大半的游戏玩家还沉浸在Tiny Wings短翅膀的飞翔喜悦中或者在所有人期待感受一下圣诞的游戏氛围时，开发者发布了不合时宜的产品可能就没有办法获得足够的前期下载量作为支撑来推动整体的营销进程。

■ 没有足够营销能力的情况下找一个优越发行商（比如Chillingo之类的）将可能事半功倍，但是这样的前提是开发者的游戏质量能够首先获得发行商的认可，如果能有效借助发行商的渠道，也许有可能会成为下一个Cut The Rope。

■ 即使自己做好了发行或者找好了可靠的发行代理，做一些类似Applicious之类的应用推荐是很不错的口碑推力，当然媒介在游戏营销中的效能也至关重要，比如Cut The Rope在洛杉矶的Hollywood & Highland Center、亚特兰大哈兹菲尔德-杰克逊国际机场的体验展示所带来的媒体关注效应。

■ 当然如果还有品牌塑造愿景的话，可以

做出类似Storm8的Story系列（最典型的是Farm Story）、Outfit7旗下的Talking Friends系列（最典型的是Talking Tom），有助于开发者挖掘用户的品牌认同度和游戏信任感（培养形成后将是非常不错的品牌概念）。

来自Getjar的数据调查做了一个很好的佐证说明，用户从哪些渠道获得应用关注，最大比例是从在线获得的占53%，真正从应用商店里获得并下载的比例仅为21.6%，来自用户口碑推荐则占到8.8%，而本文前面着力分析的第三方广告商扮演的作用则仅为7.3%（尽管已经自成产业，但是在应用的获知渠道方面并没有太超群），第五则来自社交网络，占比为5.8%。

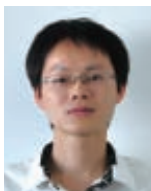
## 关于国内开发者

说起苹果对App Store对国内开发者的影响，应该不算太大。

尽管艾瑞咨询的数据认为2010年中国手机应用商店用户已经达到1400万，尽管深圳商报曾经报道称中国的苹果应用开发者已经超过3万，但仍然有一个很重要的事实：国内的作品在全球市场还缺乏足够的竞争力。

在《新京报》关于中国手机应用开发群体的报道中Frost & Sullivan中国首席顾问王煜全提到了国内目前盛行的微创新（跟风）将最终伤害创业者，因为微创新没有办法形成有效的知识产权保护门槛，也很难角逐更好的市场。

事实上，和中国社交游戏开发者走向海外面临的困惑相似，国内的iOS平台开发者同样面临着如何以西方化的思维去角逐竞争激烈但机遇无数的英文市场（在国内市场需要顾虑的是平台生态环境、用户消费习惯和付费意愿以及相关的政策瓶颈；而寻找海外市场又很难推出国际化的诸如Tiny Wings或者Angry Birds之类能够风靡全球市场的兼容性产品）。P



郑金条

游戏邦负责人，游戏邦主要关注和解析国内外社交游戏和手机游戏领域，并定期做深度行业阐述。

责任编辑：许坤铭 (xukm@csdn.net)

# Color背后：社交互动新模式

文 / 许坤铭

Color最近热度十足，原因是初创公司Color推出产品前，从红杉资本（Sequoia Capital）、贝恩资本（Bain Capital）及硅谷银行（Silicon Valley Bank）三个资本方获得4100万美元资金。这也是红杉资本至今最大一笔“推出前”投资，甚至“比给谷歌的还要多”。

在现今的IT行业里，这样一笔投资无疑是令人惊讶的，Color究竟是什么？它有哪些值得注意的特点？为何一个连正式产品都没有推出的公司能够获得以红杉资本为首的三大资本方的青睐？业界对此有什么看法？且让我们作进一步的了解。

## Color究竟是什么？

作为一个移动新应用，Color的主要功能是让用户与其他人分享自己用手机拍摄的照片，所以我们有必要先回顾一下手机拍照的历程。

1997年6月11日，来自美国硅谷的工程师Philippe Kahn为了及时与亲友分享爱女降生的一刻，将手机与数码相机结合，编写了三个简单程序，实现手机拍照并发送给他人的功能，无意间打造了世界上第一台拍照手机。不过第一款量产拍照手机是2000年夏普工程师Kazumi Saburi设计的，当年9月夏普正式发布这款内置11万像素CCD摄像头的J-SH04拍照手机。

之后夏普J-SH53和卡西欧A5403CA分别夺得了世界上第一款百万像素拍照手机和第一款200万像素拍照手机的桂冠，其后很长一段时间里，摄像头像素规格成为拍照手机的硬指标，各一线品牌也纷纷在像素数据上刷新纪录。随着拍照手机摄像头像素规格逼近卡片式数码相机，手机厂商开始认识到提升像素已不能满足消费者需要，光学变焦、视频通话、高清拍摄

图1 第一款拍照手机夏普J-SH04



逐渐成为消费者所看重的因素。

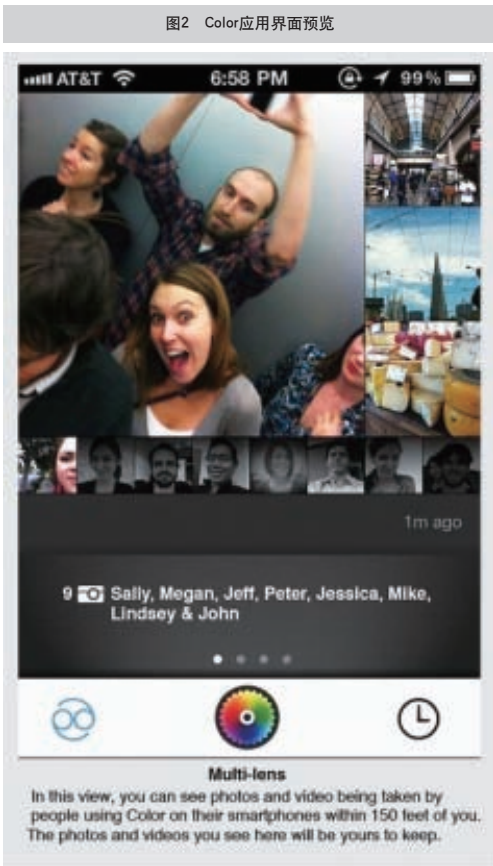
鉴于手机拍摄的照片质量日渐提升，人们对于分享照片的需求也在不断上扬。如今消费者已不再满足于用手机拍摄高质量的照片，他们更希望可以把自已拍摄的照片与其他人分享，由此催生了智能终端平台上一系列软件，Color就是其中一个代表。

但从Color的介绍中，似乎只能看出这是一款让用户与其他人分享自己所拍摄照片的程序，使用方式非常简单。用户使用手机拍摄照片，然后将照片加入到共享的照片流中，甚至没有提供照片美化的功能。如果Color的功能仅限于此，相信并不会引起红杉资本如此浓厚的兴趣，所以Color绝非那么简单，其技术内涵中必然存在着令人期待的亮点。

## “弹性网络”是亮点

如今智能手机上分享照片的软件并不少见，但是几乎都要以相对明确的朋友关系或者关注系统为基础，实现照片分享机制的前提是用户必须先与其他人建立联系，形成固化的圈子。Color在这一点上选择了截然不同的方式，用户在分享照片时，采用的是完全动态的模式，无需提前设置关注对象，只要双方都开启

图2 Color应用界面预览



了Color应用，并且物理距离满足Color的要求，就能够实现照片分享功能。

举个简单例子，当你和朋友观看篮球比赛时，只要打开手机，启动Color后拍下现场照片，对于距离拍照地点100英尺以内的人这张照片都是公开的，其他同样启动了Color服务的人就能够在照片流中看到你的照片。考虑到篮球馆拥挤的人潮，Color的照片流中会存在大量资讯，所以Color提供了分组功能，通过设定朋友名单，Color就会自动创建临时群组并且只显示群组成员所拍摄的照片，如果需要查看其他照片，用户也可以访问其他群组或者主照片流。

除了独特的照片分享机制外，Color的最大亮点在于“弹性网络（Elastic Network）”技术。在Color中用户的联系人通过缩略图列表形式呈现，不过列表中联系人排序并非用户决定，而是取决于用户与联系人之间关系的强弱，当Color侦测到用户与联系人在物理距离上的接近，就会自动调整两者关系的强度。所以Color联系人列表中，用户最亲密的家人或朋友

会排在靠前位置，相反地，如果用户长时间没有跟联系人见面，他的位置就会逐渐下调。

对于智能手机传感器的利用，Color也达到了非常充分的水平，当用户处于无信号环境时，Color会通过智能手机的摄像头感知环境亮度、麦克风侦测环境噪音等方式，确定用户位置，并将与用户接近的人划分到同一个群组。

### 打造全新社交互动模式

从应用层面上来看，Color最终目的并非只是照片分享，其创始人Bill Nguyen的说法是“创建更小的社群”。Color的团队认为，传统社交互动应用中“加好友”及“关注”的方式已经过时，如果应用本身可以实现维护社交关系的功能，用户会更加轻松。

与传统社交互动应用相比，Color可以帮用户实现更小更实时的社交圈，他们认为大众其实很愿意认识处于同一地点的陌生人，只是需要更加有效的方式。Color就是这样一种方式，通过依据地点划分用户群组的方式，Color让用户能够更轻松地开启话题并认识新朋友。

通过照片分享及基于地点的群组，Color将提供给用户完全不同的社交互动理念，围绕智能手机平台，Color让用户在不同地点加入不同社交圈，是以往社交互动应用难以实现的，打造全新的社交互动模式才是Color的真正目标。

这样的前景，正是Color能够吸引三大资本方巨额投资的原因，尽管不能断言Color一定会走向成功，但获得投资后，Color至少两年里不必担心资金及人员问题，能将精力集中在产品开发上，媒体也会在较长时间内关注Color的进展，Color团队接下来要做的就是证明自己的理念能够被大众接受，并寻求合理的盈利模式。

Color给业界带来的冲击显而易见，如果能够打造一个全新的社交互动模式，对于如今的行业格局会形成有力的冲击，届时社交互动领域将迎来新一轮的变革，目前这种趋势已初见端倪，国内也有不少开发者正致力于同类型软件的研发，在多方力量的合作下，社交互动模式的新时代不再遥远。P



# 中国版Color更给力

## ——专访“推图”创始人陈仲军

记者 / 许坤铭

美国初创公司Color获巨额投资最近成为移动领域最大的焦点，引发国内外同行的广泛关注。其实，国内也有不少同类产品，“推图”就是一个典型代表。近日，《程序员》记者专访了“推图”创始人陈仲军先生，请他讲述对Color及国内同类软件产品发展前景的看法。

### 手机拍照及社交催生推图面世

《程序员》：陈总好，能否请您用一句话讲述下，推图到底是什么？

陈仲军：“推图”是一个可以在智能终端上分享照片的网络互动交友平台。通过这一平台，用户可以实现实时拍照、滤镜处理、网络分享、地理位置定位及留言等多种功能，还能将自己的照片同步到国内主流互动社区，简单来说，就是Color+Instagram+本地化的结合体。

《程序员》：当时决定做“推图”，是出于什么样的考虑？

陈仲军：在开始做“推图”前，我们明显感觉到移动互联网未来会有巨大发展，同时手机又有着庞大的用户群体，两者结合，应该会有很大的市场空间，但具体做什么还没拿准。我尊重创新，但靠自己去创新，去摸索新应用，老实说我还输不起，不敢冒然去做小白。

因此很自然，我就把眼光放到了国外，看看国外在移动互联网上有什么好的idea，同时市场反应也不错。Instagram、Color的出现，让我看到了一个很好的机会，结合自己的判断，我认为：照片不仅具备“留存”的价值，而且它还是社交的核心基因之一，所以照片分享会是智能手机上发展SNS社交网络一个较好的切入点，做出这一判断主要依据有以下三点：



“推图”创始人陈仲军认为，“推图”是Color+Instagram+本地化的结合体

■ Flickr的图片数量有50亿张，而Facebook拥有600亿张图片，显然Facebook的社交基因里面图片就是核心之一；

■ 新浪微博里包含图片的微博所占比重越来越高；

■ QQ里添加好友之后，习惯性的第一件事往往就是查看相册，希望通过照片大概了解这个人。

想清楚之后，“推图”就这么开始了。

《程序员》：看来，您对手机拍照的发展前景非常看好？

陈仲军：是的，而且我相信未来前景会更好。因为随着智能手机摄像头的像素规格越来越高，再加上手机能随身携带这一特点，我认为在不久的将来，智能手机必将取代传统的数



码相机。

天才Philippe Kahn在1997年发明拍照手机后，到2010年全球手机拍照共达到2280亿张。随着智能手机的发展，将极大地释放大众的“拍照”需求，人们可以随时随地拍照，所以在不久的将来，大部分的拍照都会通过智能手机来完成，手机拍摄照片的数量将呈几何级数增长。最近有媒体报道，iPhone 4将打败尼康D90成为Flickr上传照片最受欢迎的相机，这也可见一斑。

所以手机将不再局限于信息消费终端，对于照片来说，手机还是专门的生产制造工具。

## Color将给社交互动模式带来变革

《程序员》：对于Color创业团队获得“推出前”巨额投资这个事件，您怎么看？

陈仲军：这件事应该说是非比寻常又理所当然。为什么说理所当然，主要有两个因素。

一方面是Color背后拥有强大的明星创业团队。比如带领团队的Bill Nguyen是业界著名的创业家，产品主管DJ Patil曾是虚拟社区网站LinkedIn的首席科学家等。

另一方面当然是Color产品本身。Color使用了一些非常有趣的技术，并创造了一个新词汇：弹性网络（Elastic Network）。Color通过算法人性化地判断用户对其他哪些用户感兴趣，

将用户进行分类，而且随着时间的推移这些联系会发生变化。Color利用了所有可用的手机感应器，如摄像头、麦克风、指南针、GPS芯片等等，智能化地依据周围光照强度、声音环境、朝向等线索判断在同一地点拍摄的照片并将用户进行分组，形成一个陌生人交友平台。

一支优秀的团队加上创新的理念，在后PC时代的萌发期将产生一批新的时代革新者，这些革新者中肯定会有下一个Google、Facebook、Twitter。

《程序员》：这是否意味照片类软件将大行其道？而在国内环境下，情况会怎样呢？

陈仲军：从行业发展的历程不难看出，这无疑是一次大胆创新。抛开“推出前”获得投资这件事，就照片分享交友社区的发展而言，未来必然会在市场中大行其道，在后PC时代占一席之地。

国内环境下，相信这类软件本地化后也将大放光彩：一如既往，国外风靡之后产品经过本地化往往会快速占领市场。从传统互联网中，不难看出交友社区的大好前景，而移动互联网的新模式也将出现一个划时代的新平台。

《程序员》：Color涉及用户个人信息、照片等，这是否会造用户隐私信息外泄的担忧？有什么办法可以有效避免用户的顾虑？

陈仲军：Color的隐私问题无疑是大问题，Facebook就曾被个人隐私问题弄得焦头烂额。

但Technology Business Research分析师Ezra Gottheil认为：“不少人，尤其是年轻人喜欢分享自己看到的東西、正在做的事情、甚至他们不喜欢的东西。也想看其他人在做什么。”

所以，我的看法是：对隐私问题，首先还是以公开开放为目的，但在产品发展过程中，我们会让用户主观地去设置自己的信息是否公开，由用户自主控制，将决定权交给用户。

## 推图发展前景值得期待

《程序员》：“推图”与同类型中文软件相比，有哪些优势或不足呢？

陈仲军：作为与Instagram同类型的照片分享应用，到目前为止，“推图”依然是国内唯一同时基于iPhone和Android两大平台的应用。

但“推图”的定位绝不是Instagram在国内的翻版，我们的定位是：基于智能手机上分享照片的网络互动交友平台，关键在于分享和陌生人交友。通俗来讲“推图”更像是Color+Instagram+本地化的结合体。

毋庸置疑，“推图”从Color、Instagram里直接借鉴了许多好的理念。例如Color理念中的即时即兴的弹性社交圈子，通过视觉日记以照片影像的方式回放当初的社交圈子，以及通过曾经圈子好友的最新社交动态（新的弹性圈子）来进一步拓展自己的社交圈，从而认识更多兴趣爱好相同的朋友；Instagram理念中照片的分享方式、滤镜处理方式、UI体验等。

这些都给“推图”很多启示，所以说Color的理念给“推图”在“近距离”交友模式带来了启发，而Instagram在通过照片分享从而找到和自己兴趣爱好相同的朋友方面，也给我们带来了启发。但“推图”更强化地理信息的标识以及标签的使用，目的就在于希望通过各个维度的属性去增强人的互动和关系维系。

当然，“推图”产品最终定位的形成，也是一个过程，在这一过程中有许多人给予我们很大的帮助，非常感谢：创新工场创始合伙人汪华、晨星创投的谢飞、蓝港在线CEO王峰、CSDN董事长蒋涛、千淘资本创始合伙人李华兵、JavaEye创始人范凯。我们一直怀着一颗感

恩的心努力把事情做好。

相对于国内一些同类产品，由于产品定位不同，不好说孰优孰劣，最终还要通过用户、市场来给出答案。

作为“推图”的负责人，我想说的是时下正值移动互联网萌发的初期，整个行业会涌现很多新idea，“推图”团队有很强的执行力而且团队成员配合默契，能够让一个点子很快变成一个优秀的产品。

《程序员》：目前“推图”用户呈现什么样的结构？用户反馈主要集中在哪些方面？

陈仲军：目前使用“推图”的iPhone用户，主要为年龄在25~40岁之间的白领，相对来说层次高一点，毕竟iPhone本身价格不菲，走的也是高端路线。

使用“推图”的Android用户，年龄相对偏年轻化多些，不过从长远看，由于Android终端价格下降、众多厂商一起发力，所以未来使用“推图”Android版本的用户会占多数，同时年轻人也更乐于拍照分享、交友。

《程序员》：“推图”是否具有有一些革命性的特点，另外对于软件的盈利模式，您有什么设想？

陈仲军：移动互联网竞争中更强调“快、极致”，所以“推图”是国内第一个实现Color社交理念（即时即兴的弹性社交圈子、视觉日记、圈子好友动态）以及Instagram拍照分享理念的产品。

而且“推图”适用的手机终端要比Color+Instagram还要多，“推图”目前能够在iPhone 3GS、iPhone 4以及Android 2.1以上的手机终端上运行。Color由于在实现方式上使用了websocket，所以导致Color只能在iPhone 4和Android 2.3以上的手机终端上运行，而Instagram目前只能在iOS平台上运行。

未来“推图”的盈利模式主要定位在滤镜效果、SNS社交虚拟物品和精准广告投放这三个方面。P



王煜全  
开放应用联盟创始人，  
Frost & Sullivan中国区  
首席顾问。

# 关于Color的三个问题

## 什么是Color？

2011年3月24日正式上线的Color引起了业界的关注。关注点有三：一是它获得了4100万美元投资，并且是在产品尚未发布前；二是公司的团队，带领团队的Bill Nguyen是著名创业家，2009年底将其创办的音乐网站LaLa以8000万美元出售给苹果，公司总裁Peter Pham曾创办过个性化开支节省顾问网站BillShrink，产品主管DJ Patil曾是风靡一时的LinkedIn的首席科学家；三是Color是基于位置信息的，从而形成社区关系的移动化照片分享应用，并支持文本和视频。

与多数社区应用不同的是，Color无需注册，没有用户名、密码，不包含个人信息，且没有采用明确的好友或关注系统，用户只需发布照片即可。所有照片都是公开的，用户可以查看和评论附近位置的其他用户最近发布的照片，可以创建群组或任意加入别人的群组参与讨论。

正如Color创始人Bill所说，“Color像是视觉日记”，向你展示周围发生了什么，你关心的人的周围发生了什么。Color的最终目标是创建一个新型、即兴、即时的移动社交网络。

## 新一代网络入口？

从互联网到移动互联网，交流更为高效、内容愈加丰富、应用日益繁荣，这是本质的趋向。然而，用户如何进入（access）网络，获得并使用各种应用，这将成为关键问题。也就是说，网络应用入口会是在什么地方？

最初，Yahoo式的浏览导航是入口；后来，以Google为代表的搜索引擎成为新的入口；如今，Facebook们打造的社交关系则是另一个入口。新兴的网络入口正逐步取代先前的，这是不

断加速的趋势。那么，未来（其实并不遥远）新的网络入口将是什么，Color是否有此潜力？对此，Color的创始人宣称，Color今后将整合Google、Groupon和Foursquare的元素。

## 如何成为Color？

把浏览导航和搜索作为网络入口的阶段，是把用户的需求作为目标点进行设计，帮助用户实现所需。到了社交网络阶段，则是把人的本质作为目标点进行设计。因其社会属性，人必然要进行社交活动，形成社交关系，一切自身需求都围绕社交关系展开。因此，基于社交关系的网络入口，是不可替代的，并且因关系的不同将有着不同的表现形式。例如，Facebook是基于双向好友关系的网络，新浪微博则体现了单向关系。Color也是一种单向的关系体现，并且更加细化，围绕一定地理位置区域内的单向关系。这是Color应用的一个关键点。

通信的发展，带动无线领域的内容不断更新。最初我们关注网络建设、通话质量，随后关注多样化应用、智能终端。因此，基于无线的应用将是今后关注的必然内容。这些内容的环节将更加细化，应用的产生、整合、推荐，最终到达用户，各环节都有很多事情可做。这显然不会像以前那样，由垄断的基础电信运营商来做，而将细化到能够抓住用户需求的企业。从这点来说，Color基于手机，结合了位置信息，依靠自身“弹性”，随着无线变化而变化，这是Color应用的另一个关键点。

纵观Color的诞生，符合未来的两个方向，一是无线应用发展，二是社区关系。结合这两个方向，有够强的团队，才能被外界看好，获得巨额投资。P



# 移动新媒介融合之道

## ——专访21世纪新媒体CTO韩磊

记者 / 杨东杰

如何应对来势汹汹的新媒介大潮，是关乎每个传统媒体存亡的重大命题。拥抱并主动变革似乎已是唯一选项，问题在于如何融合新媒介并建立成功有效的模式。

广东二十一世纪传媒股份有限公司（以下简称“21世纪传媒”）现已成为国内细分市场排名第一的专业财经商业类媒体运营企业，拥有《21世纪经济报道》、《21世纪商业评论》、《理财周报》、《福布斯中文版》、《环球企业家》等多个杂志或报纸的独家运营权，同时还与南方报业传媒集团旗下的《商务旅行》和《城市画报》建立了良好的运营合作关系。早在2007年，21世纪传媒就开始尝试各类非平面媒介的传播实践。2007年，独家经营中央人民广播电台经济之声；2008年，承“做读者的商业新闻首页”的理念推出21世纪网站；2009年，推出中国移动21财经彩信报和手机WAP网站。

随着3G和移动智能计算时代的到来，21世纪传媒旗下21世纪新媒体又于2010年初陆续推出基于iPhone、iPad和Android手机平台的移动媒体应用。截至发稿时，用户已逼近30万。相对于游戏或大众媒体应用，这个数字并不十分抢眼。但21世纪基于新媒体的探索已逐步形成一套成功的业务模式，创造出令人满意的效益。近日，《程序员》对21世纪新媒体CTO韩磊进行专访，了解21世纪传媒对移动媒介的思考和实践。

### 移动终端天然具备媒介的众多特质

《程序员》：21世纪新媒体如何看待移动互联网的发展？

韩磊：从媒体经营的眼光来看，智能移动客户端是非常有效的传播媒介。媒介的作用是让信息能够方便、快捷、低成本地生产、复制和传播，而移动媒体正好可以达到这个目的。我认为移动媒介给传统媒体带来的最大好处在于绕过发行渠道，直接和用户对话。

纸质媒体发行很麻烦，以《21世纪经济报



21世纪新媒体CTO韩磊指出，数据挖掘是新媒介融合移动业务的核心

道》为例，一份有几十上百版，携带不便，每篇文章都是独立的，也不方便网状阅读。它的发行是通过邮政和渠道商，每天分发70多万份，整个过程相当复杂。原子的速度比不上比特的速度。如果用移动媒介来做，信息数字化后，发行会更有效，成本可以大幅节省。

21世纪传媒非常看好智能手机的发展。移动互联网是能看到的最具发展潜力的领域。报纸的未来可以预期，网站的未来也可以预期，但移动互联网的发展还有无穷变化，机会也就蕴含其中。从2007年起我们就开始布局新媒介，在2009年末开始进行移动媒体方向的探索，至去年年底，在iOS、Android、Symbian和BlackBerry 4个平台上共推出10个应用，当然重点放在iOS（包括iPhone/iPad端）和Android（即将支持Android平板）平台。

《程序员》：请简单介绍21世纪新媒体在移动媒介上的产品和成绩？

韩磊：目前，我们的应用主要有三类形态。第一种是财经资讯阅读器，在手机平台上表现为“传统”的列表式应用，而在平板上则表现为具备复杂排版能力的应用。第二种是财经音频内容。基于对财经题材和读者需求的理解，21新媒体以“电台应用”形式推出财经早8点、财经看点、冬吴相对论和高端访谈4个栏目，部分内容来自与中央人民广播电台的合作栏目，部分内容来自21世纪传媒主办、承办的各类活动。最具原创性的是“财经早8点”，由专业人员播报经编辑人员精选制作的早晨财经新闻。第三种就是平面媒体的移动客户端，主要针对平板电脑。目前我们已为21世纪传媒旗下的《21世纪商业评论》，以及21世纪传媒的合作媒体《商务旅行》和《城市画报》开发出iPad应用。

从品牌的角度看，这10个应用又可归属于三个品牌。一是21世纪财经系列；二是21电台系列；三是面向海外发行的收费资讯应用Business China。其中发行量最大的是21世纪财经系列。

值得一提的是应用推出伊始，我们对手机阅读器的读者行为定位是“阅读”。不久，我们发现来自手机应用的文章读者评论，数量超

过了来自网页的读者评论，而且评论质量普遍较高。

目前21世纪新媒体的移动应用在移动端已有近30万用户，今年我们的目标是一百万。一些运营商和手机厂商主动找到我们，要求预装。这从侧面反映出优质中文应用的发展速度跟不上智能手机普及速度的现状。

## 数据挖掘是融合移动业务的核心

《程序员》：21世纪新媒体是如何融合移动新媒介并建立具有成效的业务模式的？

韩磊：从需求上说，对于传统媒体，广告客户看中的是品牌和影响力，但对新媒介，客户一定要得到实际用户的反馈，对用户数据包括基本资料和购买行为统计的要求越来越多，但我们不能直接将数据卖给客户，而是通过策划来让用户接受推广的内容。

---

21世纪报系全媒体营销模式：基于读者数据库进行数据精准选取；以平面、电台、网站、手机广告进行品牌推广；以21俱乐部为载体通过EDM与SMS按会员地区、行业、职务、学历字段进行精准营销；通过形式多样的线下活动设计，为企业产品与品牌面向最精准人群进行更优质的营销推广。

---

新媒介带来新的读者数据收集手段。符合某个具体属性的读者人群，可以清晰地按需呈现出来。数据挖掘提供了按地域、行业、职位、年龄、性别查找，使我们得以对读者进行多维度细分，从而提供符合群体个性的内容和商业服务。

通过用户数据的收集、挖掘和管理，我们建立了拥有150万高端会员的21读者俱乐部，以前，报系面向读者的服务主要集中在财经资讯传播。现在基于新媒介，21读者俱乐部可以结合读者的地域和兴趣属性，开展各类线上或线下活动。

以某汽车厂商策划的一次活动为例。我们组织了多位知名经济学家在全国十几个城市做

巡讲，定向邀请具有购买可能性的21读者俱乐部用户参加，在会议中穿插该品牌汽车的试驾活动。厂商反馈，销售效果远超商业意向更明确的试驾会。

值得说明的是，下载和使用我们的应用，并不会主动收集用户数据。用户数据主要是通过网站和活动注册来获得。这其实是一种正向循环，用户为了获取资讯和参加活动而注册，客户因为我们能邀请到目标用户而愿意支持，我们因为客户支持而能够为用户提供资讯和活动，商业信息和用户需要并不矛盾。

目前21世纪新媒体事业部共有50多人，其中研发人员只有三名。内容人员最多，他们负责网站和移动的内容制作，其次是数据部门，负责用户数据分析和商务运营。事业部去年的业务效益令人满意，未来集团将加大这个领域的投入。

《程序员》：新媒介的发展对于传统媒体业务是否有影响？

韩磊：目前新媒体和传统媒体并无太大的冲突关系。我们认为纸媒在较长时间内仍有其存在价值，新媒体和传统媒体并不是非此即彼的替代关系。

一方面，越来越多的人使用手机来阅读资讯，但并不代表他们就不买报纸杂志了。例如我爱读的《南都周刊》，一般会买一份在家里看，在外出时通过iPad上的应用阅读。

另一方面，传统媒介媒体现在还抱有一些重要优势，例如品牌、读者、内容和广告客户的认同。但是我相信这些优势都不是传统媒体能够永远占据的。未来，品牌、读者、内容和广告投放都逐渐地会从传统媒介逐渐迁移到新媒体，但是迁移还需要经历一段也许是相当漫长的过程。

## 云端交互是移动媒介创新未来

《程序员》：下一步，21世纪新媒体在移动产品方面有何计划？

韩磊：除了现有程序的更新升级之外，我们还想尝试更具创新的应用形态，融合移动设

备的感知能力以及与云端连接的能力，推出更具互动性和视觉效果的应用。目前我们的产品只是做到了大家都做到的东西，并没有独特的创新之处。

每个产品都要放在用户需求、客户的需求以及媒体价值的交集之下进行审视，回归到媒介的价值——让有价值的资讯快速、低成本地生产、复制和大规模传播。用户的实时反馈将完全改变传统媒介的单向传播模式。动态结合云端数据与客户端的实时操作信息将有极大潜力可挖。

用户数据挖掘是我们今后的重要工作，但我认为移动还只是媒介，最关键的是本身的业务模式很扎实，然后技术架构能够支撑这个模式，根据用户的需求勇于创新，这样才能在移动互联网大潮中立于不败之地。

《程序员》：对于其他媒体进军移动互联网，您有何经验分享？

韩磊：从21世纪新媒体的经验来看，主要有几点。

1. 战略还是跟风：一个媒体投入到移动媒介开发之中，首先要从战略上考虑自己是否适合移动媒介？通过移动应用来达到什么目的？如果只是随大流，简单地将内容通过手机来发放，并没有太大的实际意义。

2. 免费还是收费：21世纪新媒体的计划是免费并在很长时间内不考虑收费。这是基于现实市场和用户习惯的考量。

3. 外包还是自己研发：一开始可以投入少量资源做外包，越往后，移动媒介与业务结合得越紧密之时，自己组建研发团队恐怕是避免不了的。

移动领域已经成为传统媒体的必争之地。可以不客气地说，传统媒体几乎全都错过了网站时代，但他们不会再错过移动互联时代。印刷媒介时代成就了华光等排版技术提供商，成就了发行渠道商，移动互联时代会成就谁？我想，程序员朋友不必一味盯着应用商店的模式。通过应用商店销售赚大钱的永远是少数。在传统媒体拥抱移动媒介的过程中，还有许多机会可供挖掘。📍

# 从移动应用创作者到产品总监

## ——专访微软MVP邹建峰

记者 / 木易

2003年至今，邹建峰连续八届被评为微软MVP，早期专注Visual C++技术，后来转到Windows Phone方向，由于在Windows Phone平台上，拥有多个应用的开发经验，获邀请加入网龙公司刚起步的无线事业部，任该部门产品总监。

邹建峰一直将技术分享作为自己成长历程的重要部分。早期参与微软技术俱乐部；在网龙工作期间，他大力推动员工间的技术交流，开设Windows Phone技术论坛，成功打造学习型部门。本期微软MVP系列采访，邀请他来分享技术人员的成长经验，从产品策划角度对开发者提出建议。



微软MVP邹建峰指出，使用微软技术者更关注应用场景和最佳实践

邹建峰认为微软技术入门简单、文档详尽、使用的人也多，相比其他平台，技术方面的问题要少得多。从这个层面上讲，他们更关注使用场景和最佳实践。

作为一名技术出身的产品总监，邹建峰对技术人员成长有一套完整的体验和建议：

初学开发者要执行：对于策划（或产品设计者）所提出的功能，就要千方百计地实现而不是发表意见，初学者一定要注意这一点，因为我注意到一些组员，工作很努力，但在开发过程里涉入产品策划的工作，不断地尝试各种参数和效果，导致最后的开发周期延迟。对于产品的开发，首先要保证产品的开发周期这个基本的前提。

中级开发者要建议：有了一定的开发经验后，要主动向策划提出建议，因为你的产品经验是他们宝贵的补充，无论策划是否采纳。此外，要不断思索你的代码是否有可以改进的空间，并不一定要马上重构。在某一个版本中，如果有合适的时间时再进行适当的重构。此外，要不断和其他开发者进行交流和沟通，并多阅读他人的优秀代码以理解学习他人的设计理念。

高级开发者要专注：开发人员到了精通的时候，就要从设计开始做起，因为此时写代码并非难事，但把一件事从头到尾做得很完美就很不容易。这个时候要专注于做一件事，即设计出优美的架构，留出足够可扩展的空间和可移植的可能，一方面以供其他组员共同开发和使用，另一方面也可以防止因需求的变更而导致整个代码极度膨胀，使得产品的研发进度不可控制。P



# Android操作系统移植经验大分享(上)

文 / 钟文昌

将操作系统移植到一个陌生平台需要了解该系统的启动流程、原理及对应的程序代码，并需具备一定的开发经验。移植Android的困难在于不知道应如何开始，而且在不同平台可能会遇到不同的问题。本文将分享移植Android的核心精神：移植Android的流程。此流程适用于多数的硬件平台。基于这个流程，我已经完成其他硬件平台的移植工作，如TI OMAP 3530等。

## 概述

Android是Google踏入手机产业的试金石，其初步发展方向是提供一个开放性的软件平台，让开发者及使用者可以自行在此平台上开发或安装软件。Android的发展重点为程序库及应用，因为只有通过简单易用的UI以及多样化的软件附加功能才能使Google迅速进入手机产业并占有一席之地。本文以Android 1.0为研究及移植的主方向。

Android的软件架构采用分层设计的概念，其优点是减少各层之间的相依性、便于独立开发、容易收集问题以及调试等，如图1所示。



红色部分为操作系统。因为智能手机的功能及周边装置日趋繁杂，包含相机、蓝牙、WiFi、GPS、触屏、电池及特别为手机或移动设备所设计的电源管理系统，所以我们需要一个操作系统来管理它们。Android采用Linux作为其操作系统内核，目前支持2.6以上版本，Android正式公开的第一版原始代码Android 1.0使用Linux内核2.6.25。

绿色部分为程序库。Android采用大量开源软件作为主程序库，这也是Android的优点，因为许多开源软件都经过一定时间的检验，其稳定性及功能也都有一定的水平。黄褐色的区块主要为虚拟机。Google开发了一个支持Java语法的虚拟机：Dalvik，其特色是Register-based Virtual Machine，而非Stack-based Virtual Machine (Java Virtual Machine)，因此Dalvik可以针对其支持的平台作最佳化处理，这也是针对嵌入式系统所作的设计。

蓝色部分为应用框架及应用。为了帮助开发人员加快开发速度，Google打造了一系列的框架，以便于独立开发、维护及调试等。

## Android移植步骤及流程

我们移植的目标为建构一个可以实际运行Android的环境，包含Android的界面和基本操作，例如键盘、触摸屏等。

### 目标平台PXA270的硬件概况

Android的运行环境有其硬件限制，Google官方建议运行Android的硬件需求如下。

CPU: ARM-based以上。

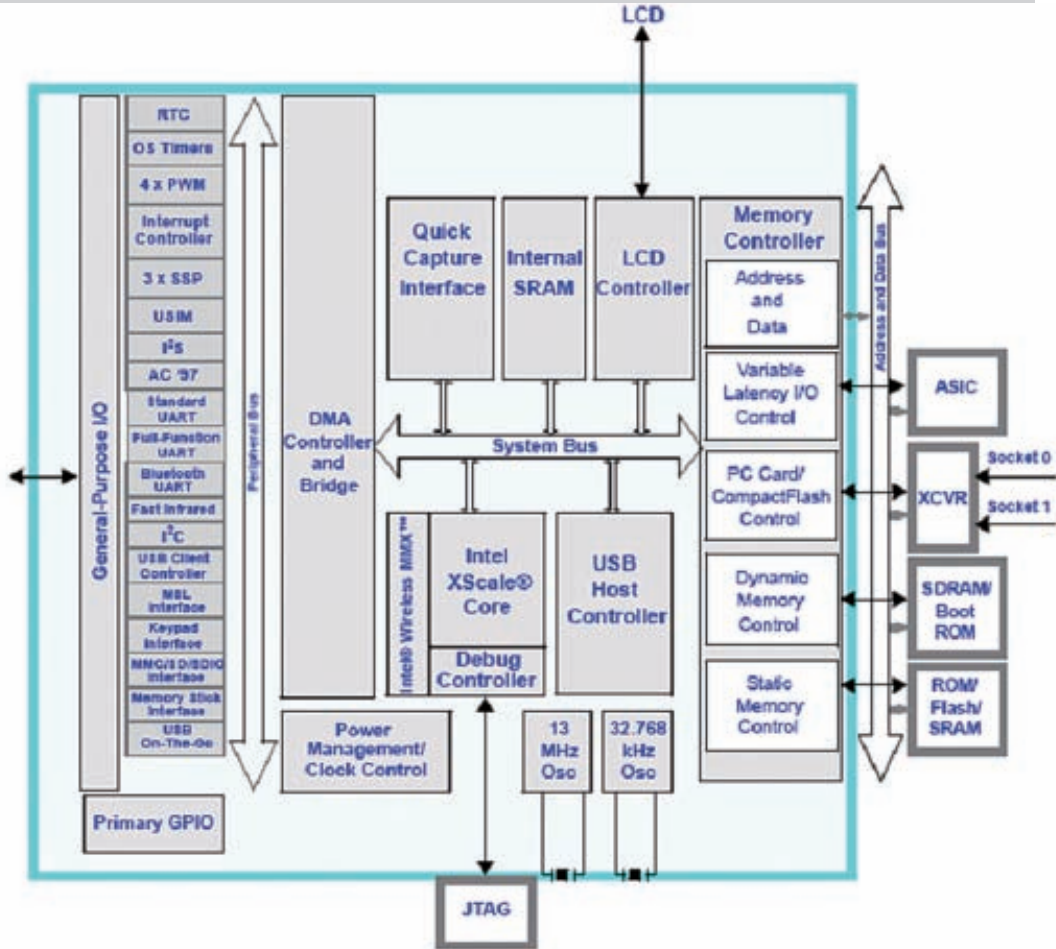
RAM: 128MB以上。

Flash: 256MB以上。

根据移植的经验，我们建议运行Android的最低硬件配置需求如下。

CPU: ARM9 Family以上，支持ARMv5以上指令集。Android Emulator使用的CPU代号为Goldfish，是ARM926EJ-S、ARMv5TEJ。

图2 PXA27x处理器功能方块图（摘录自Intel® PXA27x Processor Family Optimization Guide, August 2004. Order Number: 280004-002）



RAM: 128MB以上。我们实际移植的平台仅有64MB，因为存储器容量太小而造成Android部分应用程序在开启时需等待数秒钟，且128MB是一般智能手机的基本配置。

Flash: 128MB以上。一个基本的Android运行环境约需50~100MB，而我们实际移植的平台仅有32MB，所以我们将Android的文件系统置于外接的U盘中。

图2为本研究所移植之PXA27x的处理器内部功能模块图，表1为目标平台的硬件规格。

表1 目标平台PXA270硬件规格

CPU	Intel XScale PXA270 520MHz
Flash ROM	32M Bytes
SDRAM	64M Bytes
keypad	4x4 matrix
touch	UCB1400
LCD Module(LCM)	240 x RGB(H) x 320(V)
UART	
Ethernet	10/100 Mbps
USB	
Audio	UCB1400

准备工作

一般开发环境必须先行安装部分包才能顺利地编译Android，不同的开发平台（如Fedora、CentOS、Debian、Ubuntu等不同的Linux distribution）有不同的包名称，不在此赘述。

我们的目标是先建构一个纯嵌入式Linux系统，接着建立Android Runtime系统。以下为移植是否能顺利的关键。

细心：谨慎的态度决定移植的成功与否。

运气：包含所用的硬件是否够受欢迎、能见度是否够高、新版Linux内核是否支持设备驱动。

经验：是否有过类似的移植经验、对Linux内核、Linux设备模型、Linux 设备驱动、Linux操作环境、嵌入式系统的建置及Linux开机流程的熟练程度。

根据一般嵌入式Linux系统的软件控件及Android的软件架构图，我们将Android移植的步骤分为移植Linux内核和整合性修改两个主要的步骤，这两个步骤对应到Android架构图的关

系如图3所示。



步骤1中我们采用2008年9月的Linux内核 2.6.25 for Android 1.0 SDK, Release 1: linux-2.6.25-android-1.0\_r1。因为linux-2.6.25-android-1.0\_r1为我们整个移植的基础，所以务必谨慎小心，尽可能将硬件相关的程序代码完全移植至linux-2.6.25-android-1.0\_r1。而步骤2的部分则包含修改Linux设备驱动、Android的程序库、应用框架、Application及运行环境等。

移植Linux内核

基于Android对于Linux内核的限制，我们采用linux-2.6.25-android-1.0\_r1，而硬件厂商提供的Linux内核为linux-2.6.15.3，linux-2.6.15.3与linux-2.6.25-android-1.0\_r1的版本差异太大，所以我们使用人工patch的方式来进行移植工作。移植的原则是比较硬件厂商提供的linux-2.6.15.3与http://www.kernel.org的linux-2.6.15.3之差异。我们集中精神在硬件相关（Hardware dependent）的部分，将差异的部分加入linux-2.6.25-android-1.0\_r1中。图4为移植Linux内核的流程图。

我们可将移植Linux内核再分为下列两项子步骤。

目标为可以开机并可以看到控制台信息。

以支持前述步骤2的整合性修改为主，为搭配Android程序库及Android应用等所修改周边装置的驱动程序。

使用工具

我们使用图形化界面的比较工具进行人工移植的程序，若以Windows为开发环境，可以使用WinMerge；若以Linux为开发环境，可以使用Meld。图5为WinMerge界面，图6为Meld界面。

图4 移植Linux内核的流程图

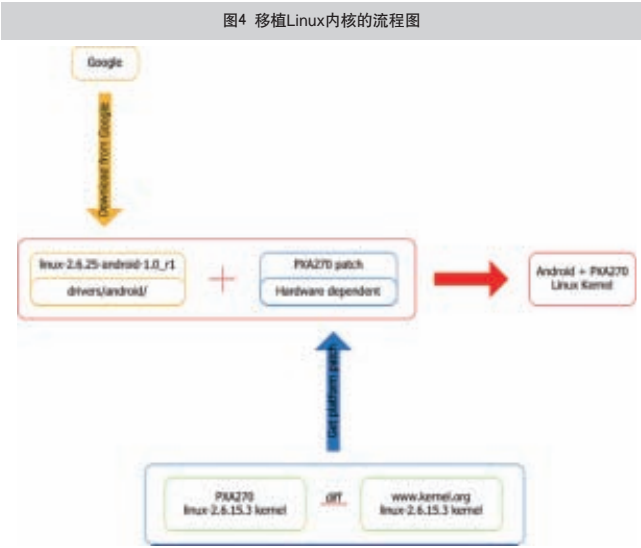


图5 WinMerge界面

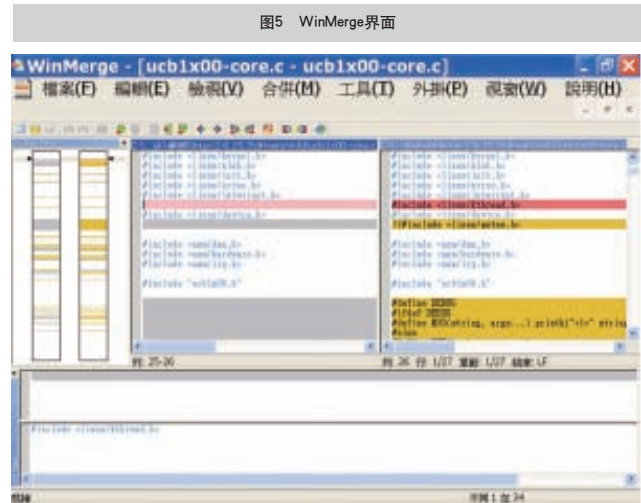


图6 Meld界面





## 可能遇到的问题及解决方法

以PXA270为例，针对新旧版本（新版为linux-2.6.25-android-1.0\_r1，旧版为linux-2.6.15.3）之间的差异而可能遇到的问题，来说明移植原则。

- 尽可能将（硬件）平台相关（包含CPU及周边硬件等设定）的程序代码从旧版本移植到新版本上。

- 针对硬件相关的程序代码，如果旧版相较于Kernel.org改动的地方（拿我们的linux-2.6.15.3与<http://www.kernel.org>的linux-2.6.15.3作比较）与新版相同，但修改的内容或方法不同，这时则以旧版为主，因为旧版是真正可以运作的版本，说明与硬件相关程序代码是没有问题的。

- 非硬件相关的程序代码，尽可能使用新版的。

- 若是遇到不知如何处理的情况，先跳过，记录文件名，再完成比较容易移植及有把握的文件，最后再回过头来处理先前跳过的文件。这部分可能需要参考部分相关原始代码才能决定要如何移植，列举如下。

参考其他同样使用PXA270的平台，比较新旧版本的差异，我们是参考Mainstone等平台的相关程序代码。

参考其他同样使用PXA270且可以在linux-2.6.25下运作的平台，观察它们所修改的程序代码。

部分Linux内核或是设备驱动的原始代码有较大幅度变动，包含以新文件名取代旧文件、资料结构改变等，这部分可能要比多个文件才能决定要如何修改，我们建议使用ctags，静态追踪原始代码所修改的部分，并谨慎地判断真正需要移植的程序代码。

上述四点是我们移植的原则。我们在移植过程中遇到的问题及解决方法可归类为下列五项。

- 资料结构改变：参考其他相同或是类似的平台，比较新旧版本的差异，我们是参考Mainstone等平台的相关程序代码。

- Init Section改变：一般发生在设备驱动上，参考其他在新版编译成功的设备驱动，研究它们从旧版本到新版本所作的修改，仿照它们，修改编译发生问题的设备驱动。

- 程序代码被分布在多个文件中：根据我们

的观察，Linux内核的众多维护者对文件命名及文件的分布位置有很严谨的考量，某些可能说不上是规范或是规定，但却都是直观且可以理解的。所以程序代码的位置在不同的版本可能有很大的变化。解决方法是只考虑新旧版间改动的部分，也就是拿我们的linux-2.6.15.3与<http://www.kernel.org>的linux-2.6.15.3作比较，将修改的部分套用至新版中，在此要注意redefine、undefine、function renamed、function prototype modified等问题，这部分要非常细心，可能要参考许多文件才能知道有哪些地方是需要修改的。

- 文件名称改变：这项一般伴随着上一项。解决方法同上。


- 如何得到新的kernel config（.config，Linux内核的设定档），有两个方法可以得到新的kernel konfig[单词写错了应该是config]。

make oldconfig：使用旧的.config产生新的.config。

make pxa270\_old\_defconfig：直接指定defconfig档，这里假设pxa270\_old\_defconfig是旧版的defconfig，将该文件复制至linux-2.6.25-android-1.0\_r1/arch/arm/configs/下。

## 编译及测试Linux内核

我们使用的toolchain为<http://www.codesourcery.com/sgpp/lite/arm/portal/package3397/public/arm-none-linux-gnueabi/arm-2008q3-41-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2>。若是编译过程中发生错误，建议针对错误消息进行相关修改，如果不知道如何修改或是该错误不影响Linux内核的开机流程，可以考虑直接关闭相关功能，目的是让移植的Linux内核可以开机并看见控制台信息。

若是运行时遇到kernel panic，尝试从错误消息中找出原因，搭配使用printk核心函数找出发生问题的程序代码；再依个人需求，尝试修改或是直接关闭该功能。（未完待续）



钟文昌（Mask）

Android顾问专家，现任深圳中一无线（www.zinn.mobi）技术顾问。中国台湾第一位公开Android的移植流程及源码作者。  
博客地址为：[www.mask.org.tw](http://www.mask.org.tw)。

责任编辑：董世晓（dongsx@csdn.net）





# 云计算应用之路

策划 / 本刊编辑部

---

云计算，作为近年来广受关注的话题，其本质是一种灵活、高效、低成本、节能的信息运作方式，借助基于互联网的一系列创新技术，存储、计算、软件、管理、网络、信息等各类资源，以服务的形式实现虚拟化、即时定制、面向领域的组合，直接满足用户的各种现实需求，真正实现IT服务的透明化。借助云计算，用户不再需要关注IT应用中底层的问题，可以专注于面向客户的业务和创新。

在全球范围内，各大电子信息厂商都在围绕云计算重新布局，主要互联网公司也在前赴后继地通过开放平台对外提供云计算服务，构建生态链。而在应用端，各行各业也意识到云计算的优势和价值，纷纷推出云计算相关应用：内部的私有云建设如火如荼，应用不断深入，并开始产生明显效益；越来越多的创新企业和创业者凭借云计算平台和服务，初始投入和运营成本都大大降低；许多新颖且富有潜力、贴近用户需求的商业模式不断涌现……

可以说，经过几年来的概念普及和技术成长，2011年，随着移动互联网、社会化网络和物联网的发展，作为整个计算系统的中枢，云计算不仅重要性日益彰显，甚至已可谓无处不在。正因如此，我们推出了本期封面报道，为您展示云计算的应用之路，可以说亮点不断，精彩纷呈。

首先，为了让大家对云计算有整体的把握和了解，我们在充分调研和分析的基础上，综述了国内云计算应用现状，并重磅推出了云计算大势图，全面、直观展示云计算的产业分布。

其次，我们做了几大采访，一方面邀请领域专家探讨云计算的发展现状和未来方向，另一方面几位云计算创业者现身说法，分享他们关于创业的思考 and 经验。

最后，我们策划了多篇内容饱满的技术文章，既包括对先进技术的解读，比如Google的Pregel、Microsoft的Dryad，也结合实战展示了云计算应用取得的突破，比如海量数据处理、存储系统、推荐系统等。

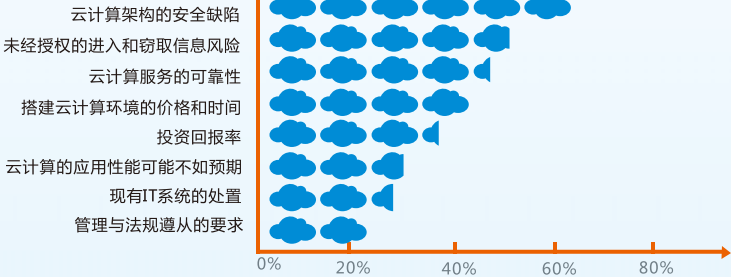
# 云计算大势图

## Big Picture of Cloud Computing

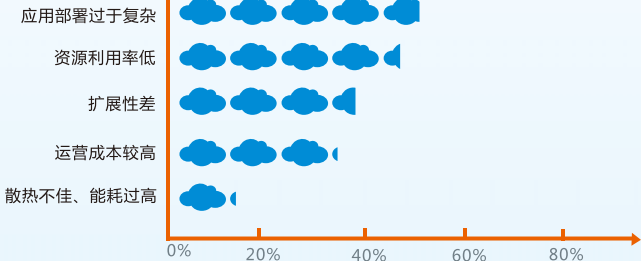
2011.5

云计算（Cloud Computing）被认为是继个人电脑、互联网之后电子信息技术领域又一次重大变革，其实质是一种基于互联网的计算模式，通过这种方式，计算资源（计算能力、存储能力、交互能力）可以动态、可伸缩且虚拟化地以服务的方式提供。这种新型的  
计算资源组织、分配和使用模式，有利于合理配置计算资源并提高其利用率，降低成本，促进节能减排，实现绿色计算。

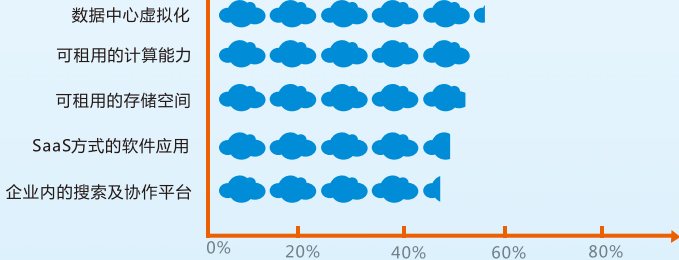
云计算存在哪些问题



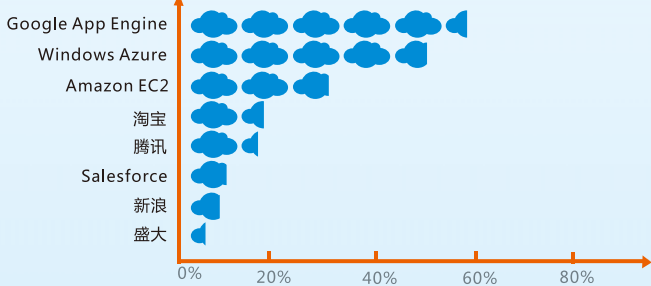
企业系统架构面临哪些问题



用户乐于接受的云计算技术



企业对云计算平台的关注度



### 云计算企业元素周期表

（不完全统计，排名不分先后）

IaaS平台 综合厂商  
SaaS平台 设备和技术提供商  
PaaS平台

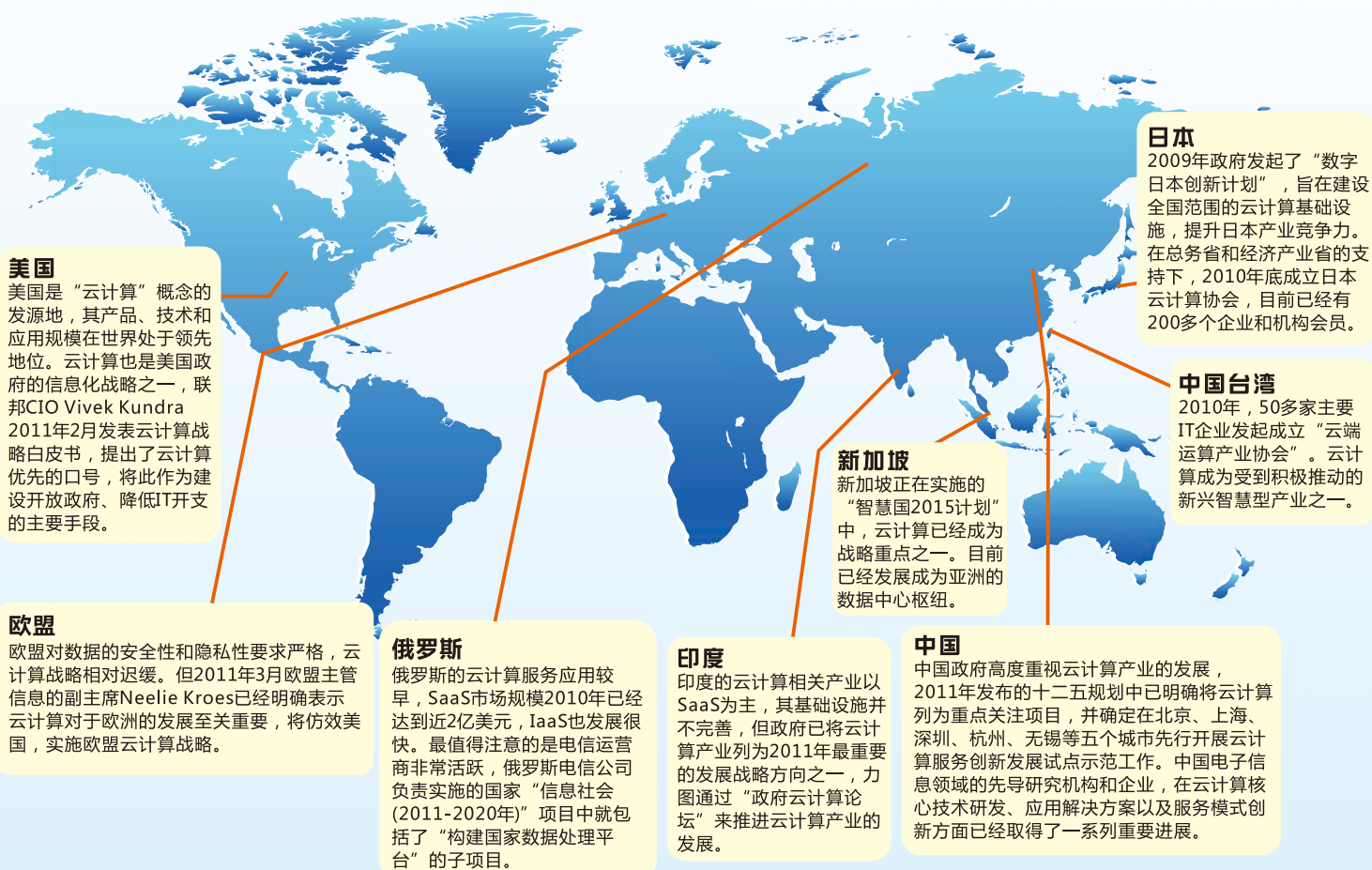
Amazon							HP
Verizon	AT&T	新浪	eBay	SAP	Zoho	华为	Oracle
RackSpace	Eucalyptus	Intuit	Engine Yard	SuccessFactors	Adobe	Intel	Cisco
RightScale	Enomal	中国移动大云	腾讯开放平台	联通“互联云”	电信“e云”	Dell	AMD
世纪互联	蓝汛	百度开放平台	淘宝开放平台	天云趋势	百会	EMC/VMware	Citrix
中国万网	阿里云	盛大开放平台	云快线	天润融通	八百客	Infortrend	Tilera
Google (PaaS, SaaS)	IBM (IaaS, PaaS, SaaS)	360开放平台	金蝶	中搜	用友	富士通	Platform
Microsoft (IaaS, PaaS, SaaS)	Salesforce (PaaS, SaaS)	中兴	10gen	Cloudera	联想	NetApp	日立
				浪潮	云海创想	友友天宇	Red Hat

### 云计算发展史 重要事件





# 世界各国和地区云计算产业概况



## 云计算技术关键词热度排行



分布式计算

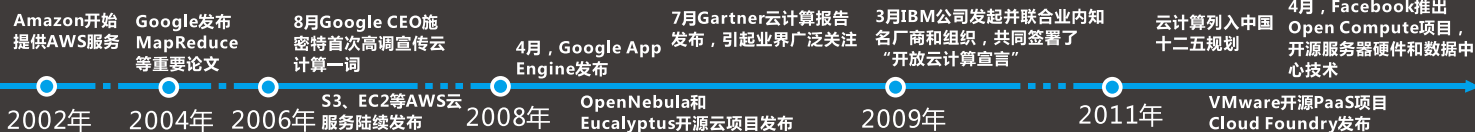
海量数据存储

虚拟化

平台

资料来源：  
CSDN云计算大事记  
CSDN云计算大调查

策划/制图 俞丽佳



# 大风起兮云飞扬 ——记2011年的中国云计算

记者 / 常政

云计算，如何落地？扣问2011年的中国云计算，这是业界最为关切的问题。近几年铺天盖地的宣传，云计算的技术体系、产业意义大家早已耳闻能详，只是它总是飘在高空，在一个“概念逻辑”里自圆其说，所以大家更关心的，是云计算能否真正脚踏中国的土壤，走出一条现实可行的道路。时值云计算产业精英的盛宴——中国第三届云计算大会召开之际，《程序员》记者采访了中国吃了第一口“云计算螃蟹”的实践者、探索者，希望他们的历程和体验，能帮助大家理解当下中国云计算产业的现实之旅。

## 转型期的机遇

从政府的角度，对云计算的重视已经达到前所未有的高度。2010年10月出台的“十二五规划”明确将包括云计算在内的新一代信息技术产业列为发展重点，而同期发改委、工信部联合发

布赛迪顾问的研究报告，中国云计算的市场规模将从2009年的92.23亿元增长到2012年的606.78亿元，年均复合增长率达87.4%。

这样的云计算大势下，自然会诞生一批弄潮者。比如国际IT巨头们，他们摸爬滚打中国市场多年，嗅觉灵敏，加上来自云计算的发源地，所以通过他们大刀阔斧般的技术市场战略，不仅可以考察他们对中国市场的理解，也可一窥中国云计算将来的局势。

2011年3月，IBM 在中国召开的年度策略发布会上，紧扣中国“十二五规划”，强调信息产业的“智慧的转型”，而云计算是他们的实施利器。记者获悉，目前IBM中国开发中心（IBM China Development Labs, CDL）已经帮助各地的中国企业和包括国家试点城市在内的众多地区实现“行业云”的落地，其中包括：电子商务云（无锡）、物流云（宁波）、金融云（杭州）、智慧城市云（北京、东营等）。

IBM全球副总裁兼中国开发中心总经理王阳表示：“行业云是我们一个非常重要的例子，中国有非常好的布局，在每一个区域、每一个政府都有自己重点行业方向，比如在宁波、长三角是物流，在苏州是制造业，在杭州是电子商务，在上海是金融，而北京则在政务等各方面有一定的重心。我们怎么把这些行业的需求用行业云的形式实现整体的提升？我们跟无锡市政府合作打造了电子商务云，跟宁波政府一起合作物流云，将来我们还会在杭州和上海打造金融和电子商务云，以及各类制造业云等。这样通过集聚的效应，我们可以在行业云上面选择最优秀应用的软件，以云的形式来提供，然后建立最标准的流程。比如在银行当中，证监会就应该对这样的流程进行考核，并通过这些形式共享资源，来提升整个行业的技术水平，这将是跨时代的飞跃。”

IBM可谓入驻中国的国外云计算厂商中的典

布的《关于做好云计算服务创新发展试点示范工作的通知》，则圈定了北京、上海、深圳、杭州、无锡这5个城市先行开展云计算服务创新试点工作，来促进云计算的落地实施。这样随着中央的支持，地方发展经济和产业升级的内在需求，目前国内已有数十个城市将云计算确定为重点发展产业，比如北京的“祥云工程”、上海的“云海计划”等。



IBM全球副总裁兼中国开发中心总经理王阳指出，云计算的集聚效应将给行业信息化带来跨时代的飞跃

型代表，他们熟悉中国政治，能从一系列重大的产业政策里敏锐地嗅出潜在的商机，并迅速抢占市场。

当然，对于外商而言，云计算技术应用的本地化也是一道门槛，对此VMware深有体会。VMware中国研发中心总经理李严冰说：“作为外商公司，代表着一些国外先进的技术。所以这些技术的本地化，是我们公司非常重要的策略。云计算牵涉很多政策性的内容，比如说你的数据中心建在哪儿，这些数据中心内部的数据和应用该符合什么样的政府规范等。所以我们希望把自己的企业打造成中国式的企业。我们正在做这几个尝试：第一，开发本地的核心技术，我们开发的技术，很多都在中国注册了知识产权。第二，从商业的角度来看，我们对中国云计算云服务将来的设想，是与中国本地的partners 结成非常深层的合作伙伴关系，这不是说简单的分销商或系统集成关系，而是深层的技术合作——在研发团队级别的合作。这样联合打造出来的产品，首先会是一个中国产品，其次它对于本地市场的需求有着很好的支持。这也是我们研发中心现在一个很重要的使命。”

## 云基地：产业链的竞争

外商纷纷抢滩中国云计算市场之际，本土弄潮者不甘示弱，他们

更理解中国国情，也不乏国际视野。宽带资本是最引人注目的一家，他们在北京市政府的支持下，花5亿巨资投入的北京云基地，其投资的10家公司涵盖了云计算产业链各个环节中的行业主导企业，包括云箱（云计算数据中心）、云计算服务器、瘦终端等硬件产品，还有云计算操作系统、虚拟化管理系统、智能知识库等软件产品，再加上视频

云应用以及系统集成服务，可以提供云计算完整的系统解决方案。

宽带资本的创始人田溯宁，相信业界不陌生，从亚信科技到网通，再到宽带产业基金，在中国IT近20多年的发展史里，始终处在风口浪尖般的最前沿。而如今他将宝押在了云计算上。原网通国际总经理、现北京云基地董事长赵安建接受本刊记者采访时说：“田总几年前就关注云计算在国外的发展，也亲自去走访了一些国外的知名企业。他对国内的IT行业一直都有很独到的判断，包括从技术的发展和需求以及云计算对未来IT、人们生活带来



北京云基地董事长赵安建认为，“共享服务模式”是基地产业链的独特之处



国内第一套正式运营使用的集装箱数据中心——云箱



的改变。因此他认为这是我们需要去做的下一件大事。”

由于10家基地云计算公司均由一家投资，所以其管理上也别具特色：据介绍，云基地专门设立了一家基地管理公司，统一管理10家公司的法律、财务、行政、人事、对外关系等事务，销售则以基地的系统集成公司天云科技为主，其他分公司主要专注于各自的业务领域。赵安建说：“这是我们这个云计算产业链里比较独到的地方，首先它使各个公司能够聚焦在业务发展；其次加强了我们的品牌凝聚力。我们将这叫共享服务模式。”

关于宽带资本将云计算产业链整体打造的原因，赵安建说：“云计算是个很大的系统工程，需要产业链每个节点方方面面的呼应。不管是市场还是技术、产品上，形成产业链般的布局，对于整个云基地产业的发展是非常有益的。特别是在云计算刚刚开始的时候，它能够集中从人才、技术以至产品各方面，具有高度的聚合互补的优势。”

赵安建认为，事实已经证明，正因为他们采取了这样的模式，对云基地产品研发和业务的发展、云计算市场的拓展，都起了非常有利的支撑作用。比如云基地已经成功地推出了国内第一套

月，云基地诞生了中国第一台云计算服务器“超云”，目前中国移动、电信、淘宝、奇虎等知名厂商已经在测试和使用；此外，云基地还创立了中国电力行业的第一个“云计算仿真实验室”。据悉，中国云计算领域的第一张订单——浦东软件园的云部署项目也由他们中标。北京云基地，正在加速“云计算落地”的进程。

## 产业需要“云思维”

北京亦庄云基地，在中国各大云计算基地里宛如一朵奇葩，不仅因为它的产品已经在逐步推广和应用，更在于它的发展模式很难被复制——因为很难找到像田溯宁这样拥有深厚的政商界资源、有科技战略眼光的领军人物。但是北京云基地产业链模式所体现的强调资源整合、形成合力的经营模式，值得业界去借鉴。

来自北京超图软件股份有限公司的副总裁王康弘将这种整合产业链上下游实现共赢的思路称为“云思维”。他说：“云计算的本质思路，实际就是共存。以电力网做比喻，电力网的存在只会让更多的电力企业活得更好。”王康弘以餐饮业为例说：“一条街如果只有一家孤零零的店，往往生意不行；相反一条车水马龙的街，两边全是餐饮店，往往都是门庭若市。地理信息产业也一样，云计算会带来一种群聚效应，使得用户们各取所需。”

但现在中国各行业解决方案厂商的思维还是比较传统，而且各自为战。这样的局面下，作为国内最大的地理信息系统平台商，超图软件的策略是先修好自己的内功，蓄势待发，为未来的云计算服务奠定根基。王康弘表示：“我们的云计算策略分两步走。第一步，先把基础软件平台进行改造，能满足第三方开发云GIS应用需要，目前这部分工作基本完成，超图软件已经先后和红帽、IBM、微软等国际著名云计算厂商合作，使得自己的核心平台能在国际主流的云计算架构上运行起来，以便大家可以在通用的云计算环境下开发自己的应用。第二步，我们也会做专门的地理服务平台，提供针对性的云GIS服务。”王康弘认为，在云计算大潮下，超图软件会从传统软件型企业走向服务型企业。王康弘说：“尽管前两年SaaS在中国的应用并不理想，但这是‘云’一直悬在空中的缘故，那时云计算的环境和技术条件还不成熟，但现在局面已经大为改观，只要提供切实满足用户需求的服务，SaaS一定会兴盛起来。”王康弘透露，今年9月，超图软件将正式对外详细颁布其在云计算领域的实质性成果。



北京超图软件股份有限公司副总裁王康弘认为，发展云计算需要有“云思维”

看来，超图软件面对云计算，走的是一种“广积粮、缓称王”的路线，而这种“未雨绸缪”般的策略在国内还有另外一种表现形式，比如新浪，立足于解决对云计算的内需，然后逐步向社会开放。

## 切入最擅长的领域

新浪研发中心平台部总监童剑接受采访时表示：“云计算是行业发展的必然趋势，所以从新浪长远发展的角度来说，是必然要切入的领域。新浪随着业务的发展和规模的扩大，本身对云计算逐渐产生了技术架构上的需求和经济管理上的需求，所以我们的战略是先立足公司内部的需求，提高公司资产的利用率和管理能力，为公司内部产品做好一个支撑，随后再去考虑商业化的可能。”

目前新浪用于内部支撑的云计算产品有SAE（Sina App Engine）、云存储，即将推出的是CDN（Content Delivery Network）。

SAE目前在业界已经有了一定的影响力，它是中国为数不多的已经投入使用的云计算产品之一，现在新浪微博上70%的第三方应用都是基于SAE。据童剑回忆，SAE的雏形诞生于2006年由新浪研发的动态应用基础服务平台，2007年在公司内部大规模使用后，他们发现无论从成本节约，还是开发效率上，均给新浪公司带来很多益处。时值云计算在西方兴起，尤其Google推出了功能类似的Google App Engine后，使他们深受鼓舞和启发，于是2009年，新浪研发团队基于原有动态应用平台的技术积累和对云计算新的理解，推出了SAE。

SAE正走在通向社会化应用的探索之路上，出于谨慎目前暂未放开注册，采用的是邀请码制度，目前已注册了1万多名开发者，运行了5000款以上的应用，其中不乏如“粉丝会”等一些颇有知名度的微博应用。童剑表示SAE今年的目标还是吸引更多的第三方应用，所以他们会致力于线下的服务资源储备、系统移植性的改进、专业技术服务等方面。至于SAE未来的商业模式，童剑表示新浪目前还没有明确

的定位，但他觉得有两种商业化思路都是可行的，一种是通过提供服务资源收费，另一种是以类似Google Market的服务形式收费。

至于云存储，新浪于2009年投入研发，去年开始小范围的试用。新浪正考虑把它做成一个存储的托管平台，目前的产品形式有微盘，而且已经开放了API，使得第三方应用访问用户在微盘里提供的内容。童剑表示，SAE、云存储在今年都将正式对外开放。

童剑说：“做云计算，建议切入企业自身擅长的一个领域，尽管应用在市场上暂时会显得不够成熟，但只要在这方面去下功夫，把最核心的产品做好，就会慢慢积累口碑，业务就能逐渐开展。以动态应用平台为例，国内尽管有新浪这样的基于PHP的APP，但还没有看到有Java的或者Ruby的，其实所有擅长这些技术的企业都可以去做。”

## 中国人的突破口

云计算的热潮无疑将带来中国信息产业的重大的重组。熟悉中国IT发展史的人清楚，变局往往意味着新的机遇，而机遇往往取决于能否掌握关键的核心技术、占据新的产业链制高点、构建完善的产业环境等方面。面对这次新的IT历史潮流，决定中国的云计算企业脱颖而出、成为世界级企业的技术瓶颈在哪里？

美国田纳西大学教授戴元顺分析道：“云计算把它分开来的话，会有很多关键点，有些关键点我们已经克服了，有些还没有。它们包括：服务器、虚拟化、网络（包括中国3G、WiFi、带宽等怎样整合到一起）、网络操作系统（即



新浪研发中心平台部总监童剑建议，切入自己最擅长的领域做云计算

样做出一个云操作系统)，然后是安全，就像怎样保证发电厂的安全，相当于保险丝这样的功能；最后一块就是终端，目前的手机终端，越来越智能，越来越像电脑，但云对于终端的需求其实是反过来，应该是越来越简单，而功能越来越强。我认为中国比较成熟的技术，在于服务器、网络，比较不成熟的地方是没有专门为云设计的终端设备，此外便是虚拟化，基本上还要用国外厂商的产品。而剩下的是两个最关键的问题，有待我们花大量精力去解决：第一，云安全，如果不解决，‘发电厂’失去效用，就没人敢来用这个电；第二，云操作系统，它涉及大片资源的调度、管理，所以也很关键。”

赵安建结合北京云基地的经验，认为目前入手云计算，需从两个层面来考虑，一是时间因素，为了争取占据市场的时间，会适当引进一些先进的技术和产品，根据市场需求再做二次开发和集成；二是要重视自主可控产品的研发，而从哪儿开始？从软件研发做起，云基地的友友系统公司正是通过推出完全自由知识产权的云操作系统及其应用，迅速在国家电力等行业得到应用，形成对国外核心技术的挑战。云基地在产品策略上正是这一思路的推动实施。

## “云政策”是当务之急

什么是中国云计算亟待解决的问题？接受记者采访的专家们，无不对云计算相关的法律、规范和标准化表达了迫切的诉求。北京云基地的赵安建说：“目前和云计算相关的政策还是照搬传统的产业政策。比如做系统集成，铁板钉钉必须有两年的业务资历，但这是云计算的核心业务，不发资质证许多项目几乎寸步难行，而目前很多拥有两年资质的企业大多数没有部署云计算服务的能力，导致很多云计算落地的目标也无从谈起。所以建议政府能在云计算相关领域的产业政策上作相应调整。”

新浪的童剑从互联网企业的角度，也有类似的困惑：“现在政府关于云计算的政策很不健全，比如对于和SAE类似的PaaS平台到底算是一个ISP，还是ICP？希望政府能及时把这个界定的框架制订出来，好让云计算厂商能清楚自己需要

提前在政策上做哪些准备，这样可以降低在政策方面的一些风险，毕竟中国在信息方面监管很严格，所以最好能尽早出台相关的政策。”

而超图软件的王康弘，则从和云计算市场销售密切相关的财会政策上表达了他的意见：“以前的软件盈利模式主要有两种：产品销售、项目服务。但现在实行云计算之后，我认为它应该是一种全新的模式，而传统的会计规则并不支持；同样对于用户，也存在很多购买障碍，比如说高校实验室，购买了云计算资源能否算实验室资产？显然现在的财务政策，并不支持这种模式，无法把你签的计算机资源租赁合同，给你换算成资产……记得过去由于软件不能算固定资产，一度导致用户们宁可去买硬件设备，而现在云计算同样带来了服务和资产关系的重新界定的问题。这都将给新兴的云计算产业带来一些挑战。期待政府能及时出台相关政策支持这些新模式。”

从这些受访专家的意见不难看出，尽管政府近年表达了对云计算的大力支持，但主要侧重在宏观口号和基础设施环境层面上，而云计算潮流带来的产业升级的背后，是传统信息产业工作模式、市场经济规则的重大变革，需要用一种全新的视角重新界定，只有及时出台一系列具体的符合云计算产业特征的法律政策和行业规范，才能使得整个产业真正健康、快速地运转起来。

据来自工信部的消息，国内的云计算标准化工作（涉及标准主体、切入点、运营、建设等方面）正在进行中，今年6月出初稿，预计年底将有正式的政策出台。

## 结束语

结束本次云计算专题采访时，记者最深切的感受是，尽管所看到的场景、产品、人，尚不足以概括中国当前云计算发展态势的全貌，但已经明显感受到了一番厉兵秣马、大潮来临前的气象。记得田溯宁曾经用“大风起兮云飞扬”来定义2011年的云计算。看来，经过几年的论证和铺垫，业界纷纷开始用实际行动表达对云计算的真实态度——2011，无疑将是中国云计算的落地之年。P



# 亲密接触中国SaaS应用

记者 / 董世晓

2006年，SaaS（Software as a Service，软件即服务）在中国落地，成为软件发展的新方向。这五年来，中国的SaaS应用情况如何？取得了哪些成果？还存在哪些问题？带着这些疑问，记者近日采访了金蝶友商网副总经理陈武强、中金数据副总经理邹均、八百客总经理李智、奥林科技总经理陈震洲，请他们分享自己的见解。

## 请评价一下，中国SaaS应用的现状是怎样的？

**陈武强：**我们研究SaaS是从2006年开始的，那时SaaS对很多公司而言还只是个概念。这个概念在2009年被炒到了顶峰，然后云计算开始走红，这对SaaS厂商是个利好消息，因为不管是SaaS还是云计算，关键都在于主机端应用，只是说法不同而已。从我们的客户量来说，服务的订阅量这几年持续快速增长。

**邹均：**我刚回国不长时间，对国内的情况并不是特别熟悉。据我所知，国外SaaS目前的发展势头很不错，对传统软件造成了挺大的冲击。从整个国际市场来看，比如像CRM增长率大约为40%，而传统的增长只有7%。而且最近这几年，国外针对SaaS公司的并购，已经达到高潮，比如说CISCO买WebEx，便是通过并购，然后再重新包装。国外市场营销类SaaS软件，有很多细分，不单单是CRM，例如我们最近看到的现金管理、金库管理等，都是通过SaaS这种方式来做。一些传统软件公司像微软，也在向SaaS、云计算这方面转型，一方面投入大量资源，另一方面把传统的Office软件等以一种SaaS的方式展现出来。就连最不积极的SAP也发布了Business ByDesign这套SaaS产品。

国内的SaaS应用大部分还是集中在CRM市场，就是说还在复制Salesforce.com模式。但大部

分SaaS厂商应该还没有找到很好的盈利模式。所以在复制Salesforce.com模式时遇到了困难，一方面与中国国情有关系，CRM的理念在国外已经很成熟，但国内还是处在从以产品为中心到以客户为中心这种理念转型的过程中；另一方面也跟这些公司的背景有关。Salesforce.com的几位创始人，之前在CRM已经有很多积累和经验，知道怎样做SaaS方面的营销。而我们中国的一些公司，实际上不是特别会做SaaS营销。

**李智：**SaaS现在处于稳步发展期，而且发展速度也在稳步提高。2008-2009年期间，诞生了不少纯忽悠的SaaS公司，它们炒概念、炒技术，无视客户的需求，把市场环境搞得乌烟瘴气。



陈武强：非常看好SaaS未来的发展



邹均：中国SaaS厂商复制Salesforce.com模式时遇到了困难

这两年随着它们的退出，SaaS的发展环境好了很多，规模也是越来越大。大家现在都默默地开发市场、做应用，踏踏实实做事，不再浮躁。当然炒作还是有，但炒的不再是概念，而是炒签约新闻、成功案例、解决方案等，已经真正把精力放在客户上面。

**陈震洲：**SaaS应该是从2007年开始在中国热起来，但中国SaaS的市场现状与中国传统软件业的现状差不多，都处于一个尴尬的局面——很多企业尤其是中小企业不愿为软件买单。中国软件市场的发育不太健康，绝大部分企业在软件领域的投入很少，还有很多在用盗版，这是一个现实。所以说纯SaaS应用在中国的现状，我认为发展得不好。

## 中国SaaS应用取得了哪些成果？还存在哪些问题？

**陈武强：**SaaS应用有以下三大成果。

第一，改变了格局。SaaS刚开始时，绝大多

数人是怀有质疑的，而现在不但有了大量的SaaS应用，还有数百万的SaaS用户，很多人已经开始接受SaaS模式。

第二，多租户架构。我们刚入手开发SaaS应用时，租户之间的数据隔离和海量数据的并发访问性能是个大问题，开发过程中曾遇到不少挑战。后来我们总结经验，推出了多租户架构，其最大的优越性在于，它从代码层面屏蔽了SaaS应用与普通应用的差异，用户只要按照不同应用的模式开发即可，完全不用考虑数据隔离，也不用关心海量数据带来的性能问题。

第三，SaaS平台。这个平台不同于手机应用商店，更像是移动运营商的BOSS。我们借鉴了三户模型，在此基础上根据SaaS的实际情况，解决了服务租赁所特有的一些问题，例如多服务实例、服务试用、服务共享等。可以这么说，在SaaS服务平台领域，完全是根据实际业务场景摸索出来的，没有捷径可走。

但与此同时，SaaS应用也存在两个问题。

第一，市场培育期长。与个人互联网应用相比，企业级SaaS应用需要更长的时间才能被用户接受。其主要原因是企业业务相对复杂，不像个人互联网应用可以快速传播，这方面需要SaaS应用提供商作出更大努力。

第二，兼容性问题。服务与产品不一样，产品只需要对自己的行为负责，但一个SaaS服务，从它上线开始就会有升级，尤其对复杂的SaaS服务，而每一次升级都需要向下兼容。

**邹均：**总的来说，SaaS取得了很大的成果，它创造了一个可以跟传统许可权制度竞争的商业模式。实际上，SaaS是从ASP（Application Service Provider，应用服务提供商）发展过来的，在多用户支持、移动虚拟化、基础设施等几个方面克服了ASP的一些局限，在市场的覆盖广度方面具备传统模式无法比拟的优势。

SaaS存在的问题主要在于运营体系、知识体系和大环境。首先，与传统软件应用不同，SaaS是一种服务，服务肯定要有运营的支撑体系，而且跟传统的营销模式又不一样，对客户需求的管理，有它自己的特点；其次，如何把传统的东西变成合适的SaaS模式是必须要探讨的，如果没有创新，只是简单地把传统软

件变成SaaS的话，就没有一套相对应的组织架构、人员技能，还会产生与客户使用习惯不匹配的问题；此外，国内的带宽跟国外相比还是比较有限的，而且价格昂贵，这限制了SaaS的普及。

**李智：**SaaS应用方面有很多成功的案例，主要包括销售自动化、客服流程管理、协同办公等。一提SaaS，早期大家往往认为是一个简单化处理，以为就是一个OA。我觉得从实际意义上理解SaaS的话，它应该是大型管理软件的升级版，是将大型管理软件而不是一个工具类的软件搬到互联网上。现在很多公司推的SaaS产品是面向中小企业的，但我觉得真正的企业应用，还是需要稍微大一点的企业才能用，因为太小的企业如果用SaaS做管理的话，反而是个累赘，还不如用其他管理类软件效率高。

**陈震洲：**这方面的代表性成果有很多。以我们为例，之所以选择国际物流这个行业，因为它符合SaaS物联网模式。国际物流使用传统软件有很大的问题，因为它本身是一个需要跟同行、客户，包括海外目的港的物流公司进行互动的行业，具备互联网互动的特点。如果企业能提供一套基于互联网SaaS模式的应用，那么这个行业里面的客户就愿意把以前的软件扔掉，从而为SaaS买单。与通用市场相比，行业市场传播会比较快，因为本行业里的企业之间都比较熟悉（相当于一个社区，我们称之为商业社区），而且这些企业都集中在中国的港口城市，因此在这样一个垂直的商业社区里面进行推广的成本比较低。此外，通过商业社区，可以在同一个行业里面，产生多个可盈利的增值业务。

但是，要想切入行业，必须具备一些行业背景，对这个行业要有很深的了解，或者有能够经营这个行业的团队。

### 当前SaaS还有哪些技术难点需要解决？

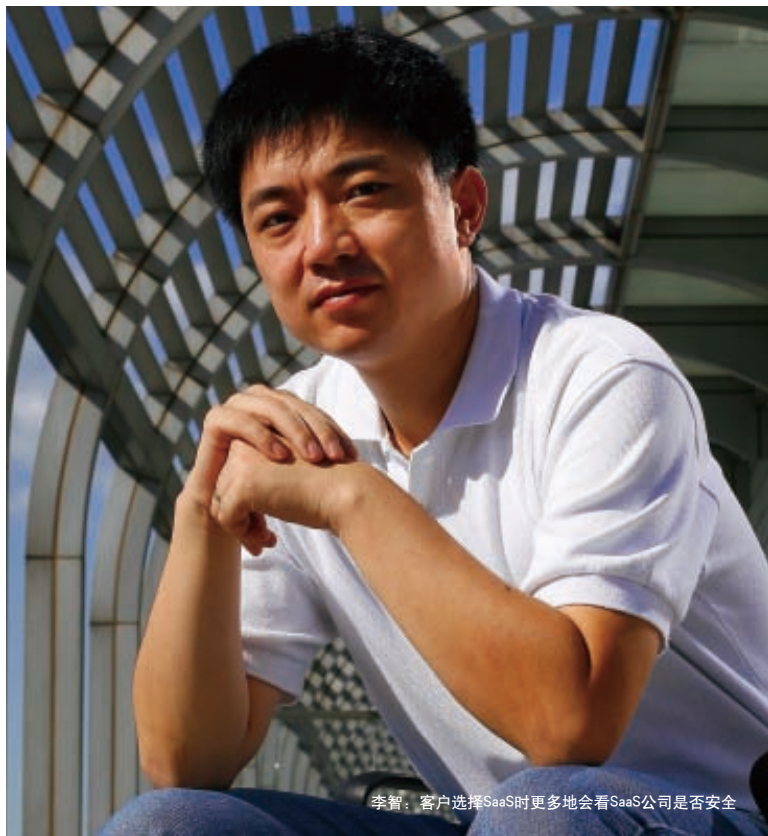
**陈武强：**对于我们来说，开发一个SaaS应用和开发一个传统应用是完全一样的，开发服务本身没有难点。就像前面提到的，SaaS的升级和集成是我们认为的难点，升级的难点

在于合理控制需求边界，集成的难点在于降低成本。

SaaS服务不是孤立的，不同的服务之间存在集成需求，甚至还会和传统软件产生集成需求。集成的难点在于：SaaS用户都是低端用户，服务与服务之间的集成订购过程必须足够简单，这涉及成本问题和技术门槛问题，用户期望的是点一下鼠标，但实际上涉及到太多参数，要做到“简单”很有难度。

每个SaaS服务上有成千上万的服务实例，集成时要考虑身份问题。为了实现SaaS集成，我们专门研发了数据交换中心，其本质是一个SaaS化的数据总线，并在此基础上实现了一个能与ERP集成的在线订货系统。虽然我们做到了，但实现这种集成的成本很高，要把这个事情变得容易还有很长的路要走。

**邹均：**第一是标准化，标准涉及很多方面，目前并不统一，比如不同的SaaS之间怎样整合、怎样与传统应用或者企业内部的应用连接整合等，还有很多问题有待解决。第二是安全，这是大家都比较忧虑的事情。对SaaS来说，数据的归



李智：客户选择SaaS时更多地会看SaaS公司是否安全



属权在法规上有一些不确定性，这实际上也是有待解决的难点。因为SaaS是一种服务，那么它就必然涉及服务合同中问责的问题，如果客户用的SaaS出了问题怎么办？在使用传统软件时，公司内部一般有明确的问责机制，通过人工方式来建立合同和解决问题。但现在是通过互联网以网络的形式来使用服务，这方面的服务合同如何制订、怎样监管它的执行？我觉得也有些技术难点要解决。

**李智：**在技术方面，SaaS已经比较成熟。我想谈的是数据安全的问题。从整个行业来说，大型企业对SaaS的态度还是谨慎的，主要担心数据安全的问题，不愿意把自己最核心的技术、敏感数据等放到互联网上。客户在选择SaaS时，更关心的还是你整个公司的形象，包括公司的实力。客户不会简简单单看你的系统是否安全，更看重的还是看你整个公司是否安全。

**陈震洲：**SaaS是有很大的技术门槛的，很多公司把SaaS想得太简单，比如之前做传统软件的话，如果需要个性化，可以在本地完成。但如果是SaaS模式，将如此多的客户放在同一个平台上面运营，个性化就是一个很大的问题。如果真正

能把SaaS做大成商业化运营的模式，没有两把刷子是不行的。在技术方面多进行投入，才有可能让客户个性化的需求快速得到响应，运营成本能够降到最低，我觉得这是中国很多SaaS公司面临的技术难点。

## 从传统的架构迁移到SaaS，有哪些需要注意的问题？

**陈武强：**在迁移的过程中，有三个不容忽视的问题。

第一，租户数据隔离。使用多租户技术可以有效规避此问题，而如果自己从头做起的话非常困难。

第二，服务端性能问题。切忌把长周期的运算直接搬到服务端实现，一个能够承受海量用户的SaaS服务，每个请求完成时间必须小于3秒。如果一定超过这个时间，又必须放在服务端，那么就必须考虑采用一些对策。

第三，传统软件在局域网中与服务器端通信通常是不考虑流量的，但SaaS应用要通过互联网订阅使用，互联网的带宽很昂贵，省带宽就是省钱。

**邹均：**SaaS的业务模式与传统模式不太一样，分为企业级和消费级。

企业级SaaS强调的是盈利模式，通过什么方式能够做大。如果单单对某一家企业，可能传统方式的优势比较强。在SaaS这方面做的话，肯定要有一定的规模。如果形成了一定规模的话，那问题就来了。因为这些企业有不同的业务流程，至于如何能做到一个可以支撑不同业务的SaaS，那就要回到如何配置来灵活支持不同企业的问题上，这还是有些难度的。另外，企业级的SaaS应用，多数还是跟事务处理有关，所以对数据的一致性要求较高。SaaS模式如果要做成规模，实际上用到云计算的分布式处理。传统的关系型数据库对这种事务处理的支持并不是特别好，所以传统的数据库应用搬到SaaS，还是会有很多的技术困难和挑战。

消费级SaaS实际上与创新和营销方式有关，



陈震洲：SaaS是下一个十年软件业的发展方向

因为这种模式一般都是要免费，这就涉及怎样做到交叉经营、怎样用一块收入来补贴这种免费的SaaS，这也是很大的挑战。很多时候是要通过创新、通过提供真正有价值的一些应用来赢得客户。

**李智：**至于迁移的注意事项，我觉得没什么太多可以说的，迁移的过程只是一件很自然的、水到渠成的事情。一个更有意义的现象是：很多客户端迁移并不只是简单地把老系统转到我们的系统上来，而是会尝试实现一些新功能。事实上，他们往往是因为有新的需求才迁移的，因为像老系统做二次开发一般都需要一些费用，而且费用还非常高。而通过向SaaS迁移，他就可以用一个比较低的成本来获得一些新的应用。

**陈震洲：**就我们的客户来说，他们把业务迁移到SaaS，一般都要放弃原有的软件。我们会把他们的数据做一个迁移，帮他们导入原来的数据，但是大部分数据不需要迁移，可以直接用。

## 中国SaaS的未来路在何方？

**陈武强：**从概念上来说，SaaS不是新鲜事物，早已经被广泛使用，电子邮箱就是最典型的应用。客户是感性的，他们使用一个应用的要点是：需要、服务好、安全、易用、低成本，而不是SaaS并不是关键因素。因此，不是所有软件适合通过SaaS模式交付，SaaS应用如果能够抓准需求点、提供良好的服务并取得用户的信任基本就成功了。

我非常看好SaaS未来的发展。经过前几年的积累，我们已经拥有了技术基础、成熟应用和稳定的客户群。同时，借助当前云计算的热潮，SaaS服务厂商能够获得更好的发展环境，在云计算时代，我们相当于已经先行一步。

**邹均：**首先，我觉得一个自动化的业务流程外包的模式，可以给客户提供比较高的价值。其次，就是数据安全的问题，如何给客户提供数据安全的保证，这方面可能与一些监管环境有关，也跟相关安全技术有关系。再次，是运营环境，SaaS实际上是一种服务。现在很多时候，我们是

从电信这种传统的业务支撑系统来考虑，怎样能够找到一个比较适合SaaS和云计算的环境，而不像这种笨重的电信运营架构，相信将来在这方面会有一些突破。最后，就是生态链的问题，这值得国内SaaS厂商进行探讨。通过生态链的整合，能够充分发挥不同厂商的优势。目前国内SaaS厂商很多时候都是用一些传统的方式来开发，然后把它变成SaaS，而并不是在整个云计算环境里面做开发、测试、运行和维护，这样做很容易出现问题。所以依托PaaS厂商提供的开发环境进行SaaS开发，才是正道。

**李智：**SaaS近几年的发展，符合这样一条曲线：刚开始时增长较快，然后进入炒作期；接着泡沫破裂，进入低谷期；最后从低谷弹起，慢慢往上升。可以说，SaaS已经处于一个稳步发展的普及、上升阶段，并且会持续下去。

**陈震洲：**SaaS是我们下一个十年软件业的发展方向，也是软件业转向了一个运营和服务的模式。既然软件在中国没有价值的话，那么软件的价值就在于服务。一个企业，它不太愿意花十万、一百万买软件，但是如果软件能在管理上面帮到企业，能够让它降低成本、提高效率，那么还是有很多企业愿意来买单的。再加之如果一个公司的信誉、安全上能够让客户打消疑虑，那么这种服务的模式是能够被客户接受的，这个事实已经被很多企业证实了。此外，SaaS模式不存在盗版，也不存在应收款的问题，如果能快速发展客户的话，今后的盈利模式、收费模式也会相对稳定，与传统软件相比还是有优势的，这也说明了为什么还有这么多的公司坚持做SaaS，就是这个道理。

今后SaaS的发展方向，从我们的角度来讲，是跟做生意结合在一起。软件本身是个软生命，你需要把它做成模式迁移到SaaS里面去，让你的客户能够做一些生意上的交易互动，也就是向电子商务的方向发展。把客户集中到这个SaaS平台上面，通过将管生意跟接生意无缝对接，成本才能降低。另外，平台在交易过程中相当于一个裁判，保证平台上面交易客户的信用，同时也是对他们服务的一个保证，如果有什么问题，平台可以出来协调。P

# 看云卷云舒——各公司云计算之路漫谈

■ 文 / 陆嘉恒 许翔

当今IT界风起云涌，云计算成为炙手可热的话题。在科研人员眼中，云计算或许还称不上是一种新的技术，它只是分布式系统的发展，是并行计算、分布计算、效用计算的结合和演化。但是，云计算拥有优秀的海量数据存储和处理能力并能够带来与以往不同的用户体验，因此受到各大公司的青睐。各公司纷纷在云计算上部署业务并推出云计算产品（表1）。

EC2主要通过虚拟化技术向用户提供计算资源，而用户可以根据自己的需求动态地申请或停止资源的使用。随着并行处理和MapReduce编程模型的流行，Amazon也推出了Amazon Elastic MapReduce，它是通过在Amazon EC2和Amazon S3上构建Hadoop框架而实现的，这使得用户可以方便、高效地并行处理海量数据。

存储服务主要是Amazon Simple Storage

表1 各种云计算产品				
公司	主要业务	云计算应用情况	主要云计算产品	产品类型
Amazon	B2C在线零售业务	自主应用、IT运营、财务应用正逐步迁移到云计算上	AWS（面向用户）	IaaS PaaS
Yahoo!	搜索引擎、新闻资讯、邮箱、社区等	主要使用Hadoop、Sherpa和MOBStor进行数据处理和存储	PNUTS（用于业务支持）等	PaaS
eBay	线上拍卖购物网站	基于Hadoop构建自己集群处理存储和处理海量数据	Turmeric（开源）	PaaS
百度	搜索引擎	使用云计算进行网页数据挖掘、聚类并对日志存储分析	框计算	PaaS SaaS
IBM	IT设施制造、管理和服务咨询	决策分析、工作协作、桌面云、存储云、数据中心整合等IBM6+1云计算解决方案	Tivoli Service Automation Manager、Rational Application Developer、LotusLive SmartCloud等	IaaS PaaS SaaS
Google	搜索引擎	推出Google文件系统、BigTable技术和MapReduce技术	Google App Engine、Google Docs等	IaaS PaaS SaaS

下面我们介绍几个具有代表性的公司，看看各大公司它们是如何分享云计算这道大餐以及如何使用云计算来支持自己的业务的。



Amazon是最早涉足云计算的公司之一，其云计算产品主要集中在基础设施层和平台层，并以Web的方式向用户提供服务，用户使用其计算、存储等功能时无需考虑数据的具体管理和维护，同时按使用付费。

AWS核心服务主要包括计算服务、存储服务、数据库服务和消息传送服务。

计算服务包括Amazon Elastic Computing Cloud（EC2）和Amazon Elastic MapReduce。

Service（S3），用户通过S3将数据存储存储在云上，其存储容量是可以动态扩展的。用户通过网络就可以获取自己的数据，同时用户部署在AWS云上的其他服务也可以直接访问S3上的数据。

数据库服务主要包括Amazon SimpleDB和Amazon Relational Database Service（RDS）。其中，SimpleDB主要是基于S3和EC2向用户提供轻量级的数据存储和查询服务，且无需规定数据的模式，因而关系数据库中的数据难以迁移到SimpleDB上。因此，Amazon提供关系型数据库RDS服务，用户通过RDS可以使用MySQL所具有的功能，以后还可以使用Oracle 11g。

消息传送服务包括Amazon Simple Queue Service（SQS）、Amazon Simple Notification Service（SNS）和Amazon Simple Email Service



(SES)。SQS提供计算机之间传递和存储消息服务，它使得在Web服务之间构建工作流变得容易；SNS则是一种在云中建立、操作和发送通知的Web服务；SES提供高扩展的大量事务性邮件发送服务。

同时，AWS还提供与核心服务相关及其他的一些服务，如AWS Elastic Beanstalk可以让用户方便快速地在AWS云中部署和管理应用程序，Amazon Virtual Private Cloud 使得企业通过VPN将自己现有的基础设施连接到AWS的计算资源，从而提高企业在安全服务、入侵检测等方面的管理能力。

如今已经有很多客户在使用Amazon的基础云服务，其中包括著名的互联网公司和创业型公司。2010年，AWS部门的收入已达5亿美元，已经成为Amazon收入的重要组成部分，Amazon的AWS在基础设施云中占主导地位。



Yahoo!是较早将业务部署到云平台上的公司，如今Yahoo!云计算平台上主要有三大利器：Hadoop、Sherpa和MOBStor。

Hadoop作为开源的云平台得到很多公司的应用，包括Facebook、Twitter、百度等。Yahoo!现在已有近百人的团队在开发Hadoop及其相关项目，且在技术研究方面处于领先地位。Yahoo!将Hadoop应用于自己的各项业务和产品中，包括数据分析、内容优化、反垃圾邮件系统、广告的选择、大数据处理、用户兴趣预测、搜索排名、广告定位等，例如实时服务系统从数据库中读取用户到兴趣的映射，Hadoop集群则基于最新数据重新排列内容并更新页面。

Sherpa是Yahoo!的分布式数据存储和服务平台，主要针对的是结构化的记录型数据。Sherpa平台上最关键的是PNUTS——一个大规模并行处理的分布式数据库系统。PNUTS实现了地理位置上的分布存储，并根据用户的地理位置提供快速的请求响应。PNUTS放弃了传统关系型数据库强一致性的要求，虽然保证了数据最终是一致的但也存在记录不一致的情

况。PNUTS是由Storage Units、Router、Tablet Controller和Message Broker四部分组成。Storage Units (SU) 是具体的物理存储器，Routers则为客户提供数据存储的具体位置，Tablet Controller控制数据的存储，而Message Broker通过使用Yahoo!基于主题的发布/订阅系统来传递数据的修改信息。

MOBStor主要用来存储非结构化的二进制大文件如图片、音乐、视频等。MOBStor在对数据提供可靠、安全存储的同时还能向用户提供快速的响应，并使用REST传送数据。MOBStor的体系结构为三层，分别为对象存储层 (Object Store Layer)、本地对象管理层 (Local Object Management Layer)、全局对象存储层 (Global Object Management Layer)。数据在存储层上抽象为ObjectStore，并通过本地URL来获取数据，因此储存层保存着本地URL到物理储存的映射。本地数据管理层完成数据复制、元数据管理、获取权限控制等功能，而全局对象管理层提供REST API、授权服务、管理请求路由等一些面向用户的功能。

Yahoo!一直推动着Hadoop的发展，Hadoop以开源且优秀的性能受到许多商家的青睐，学术界对此也保持着较高的热情，最近几年举办的Hadoop中国云计算大会也受到许多人的关注，并成为国内Hadoop爱好者交流的平台。



eBay是全球知名的向个人和企业提供销售商品及相关服务的在线交易平台和购物网站。面对数亿商品的信息，eBay采用云计算平台来存储和处理海量数据。

eBay基于开源云计算框架Hadoop建立了自己的集群——Athena，它分为五层。

■ 核心层，包括Hadoop运行时环境、通用工具和HDFS，其中文件系统为读写大块数据而做了一些优化，如将块的大小由128MB改为256MB。

■ MapReduce层，为开发和执行任务提供API和控件。



■ 数据获取层，数据获取层的主要框架是HBase、Pig和Hive。

■ 工具、加载库层，主要的加载库有：统计库（R）、机器学习库（Mahout）、数学相关库（Hama）和eBay自己开发的解析网络日志的库（Mobius）。

■ 监视和警告层，Ganglia是分布式集群的监视系统，Nagios则用来警告关键事件如服务器不可达、硬盘已满等。

这个集群由多个团队共同使用，因此采用Hadoop公平调度器（Fair Scheduler）来管理、分配团队的任务池并分配权限、设置优先权等。

最近，eBay推出自己开发的开源云平台项目——Turmeric，这是个面向服务构架的综合平台，用户可以在上面开发、部署、管理和监控服务。该平台主要包括以下几个方面。

■ 核心运行库，基于流水线架构，主要用来运行服务和客户端。

■ 开发工具，提供Eclipse插件来帮助创建服务和客户端。

■ 监控器，该监控器包含多个组件：收集客户端和服务端信息的运行收集组件、数据配置组件、监控节点信息的服务监控组件以及方便用户查看的控制台组件。

■ 安全服务，提供平台上的安全服务（如验证、授权、组服务等）与策略服务交互，其中策略遵从XACML（可扩展的访问控制标记语言）结构和语法。

■ 策略管理控制台，管理策略的定制。

■ 仓库服务，这是端到端平台上服务注册、附件管理功能的抽象，具体的功能依赖于下层的仓库产品。

■ WSDL声明服务，通过这项功能用户可以定义服务的WSDL接口。

由于Turmeric是开放、可扩展的平台，用户可以增添或替换平台上的服务和功能以满足自己特定的需求。

基于公司规模及业务需求，eBay积极使用云计算通过构建自己的集群系统来处理大规模的商品数据和用户需求，面对数据的增长云计算成为互联网公司的首选。



百度作为以搜索引擎为主的公司，对海量数据处理的要求是比较高的，需要在线下对数据进行分析，还要在规定的时间内处理完并反馈到平台上，这就需要通过性能较好的云平台进行了，百度使用Hadoop来处理大规模数据。在百度，Hadoop主要应用于以下几个方面。

■ 日志的存储和统计。

■ 网页数据的分析和挖掘。

■ 商业分析，如用户的行为、广告关注度。

■ 在线数据的反馈，及时得到在线广告的点击情况。

■ 用户网页的聚类，分析用户的推荐度及用户之间的关联度。

同时，百度推出了与云计算类似的框计算，即基于互联网的一站式服务。用户在框中输入服务需求，系统能够识别用户的需求并基于服务提供商直接返回服务，而不是返回相关服务的链接。框计算的背后需要复杂的需求分析，以及对各种服务的集成。准确的说，百度提出的框计算应该是云计算的一种应用，但是框计算确定用户需求后会选择哪个服务提供商为用户提供服务，百度的选择甚至影响一些服务提供商的发展，这说明云计算带来的影响不仅限于技术，任何互联网公司都不能置身事外。



IBM是涉及计算机软硬件制造、销售及IT服务的跨国公司，如何整合自身的全球资源提供优质服务，IBM使用云计算解决方案。业务决策需要海量数据的存储和分析，IBM使用“蓝色洞察力”决策分析云系统来处理全球小型交易等业务。公司内数万员工需要配合工作，IBM利用LotusLive使得公司内员工及合作伙伴的协作高效化。同时，IBM使用桌面云实现了桌面的集中虚拟化，提高了控制管理的安全性及维护的方便性。存储云实现虚拟化储存，将数据储存在一起，有效地节约了空间。

最近几年，IBM将分布全球的数千个数据中心整合成几个，里面的应用程序也随之部署在云计算上，实现资源整合。此外，IBM也将云应用在生产研发中，如在芯片设计时需要昂贵的软件和巨大的计算存储资源，采用设计制造云后既减少了设计成本提高资源利用，同时也保证了知识产权的安全。

IBM在云计算的产品主要涉及云计算环境的搭建和云计算解决方案的提供。IBM在虚拟化和云计算方面拥有众多产品，涉及基础设施层、平台层和应用层。IBM Ensembles将计算、存储、网络等物理资源虚拟成基础设施资源池，隐藏了内部实现细节向上层提供资源访问和管理的接口。IBM TSAM提供应用服务的管理平台，包括服务的设计、部署和运行管理整个阶段，实现了硬件到操作系统、中间件的整体管理自动化。IBM SmartCloud是用于数据中心的云服务软件，在向公司用户提供云存储的同时也提供数据分析、销售管理、测试工具等服务。基于对云计算的认识，IBM提供公有云、私有云和混合云多个选项，减少用户对隐私、安全的顾虑来吸引用户。



Google是涉及搜索引擎、云计算的跨国公司，其在云计算的技术和产品都处于领先地位。Google在云计算拥有著名的GFS、Bigtable、MapReduce三大技术。GFS是针对数据密集型应用的分布式文件系统，它可以运行在廉价硬件环境上并拥有较高的容错性，系统使用主控服务器向用户提供元数据信息和数据存储位置，而块服务器提供数据操作。Bigtable是管理海量结构化数据的分布式存储系统，它提供简单的数据模式动态地控制数据部署，通过维护稀疏的多维排序映射表来检索数据，系统中主服务器分配表块而表块服务器主要管理表块集合。MapReduce编程模型利用输入键/值对集输出新的键/值对集，其将并行计算细节隐藏，用户只需实现map函数和reduce函数即可。

Google云计算产品主要有Google App

Engine和云端应用程序。Google App Engine是一种架构Web应用程序的平台，用户可以利用其提供的SDK开发Web应用程序，并将其上传到平台上，Google负责应用的管理和维护，目前支持Java和Python两种语言开发。同时，平台还提供网页抓取、图像操作、邮件等多种服务。Google提供很多云应用程序，其中Google Docs使用较广，它是基于Web的文档处理程序，支持在线编辑和多人协作，并能够保存和修改文档内容。Google Cloud Connect可以直接将本地Microsoft office和云端的Google Docs联结起来，文档在本地保存的同时也保存在云中，并通过链接向别人分享自己的文档。Google以其先进的技术云计算发展中处于领先地位。

### 小结

互联网上数据迅速增长，分析和处理这些海量数据成为互联网公司业务的迫切需求和重要挑战，互联网公司因此成为云计算的主要提供者和使用者，以及重要推动者。云计算在提供强大性能的同时也带来新的商业模式，并作为新的盈利点成为各大IT公司的必争之地。普通用户也会从云计算发展中受益，享受到高性能所带来的服务和便利。在云计算的发展和推广过程中也存在很多问题，其中安全性和标准化一直是人们担心的：在云中如何保护用户的隐私并让用户放心？各大IT公司又能否推出统一的标准？这些问题直接影响着云计算未来的发展情况，不管怎样，云计算所代表的发展趋势还是毋庸置疑的。P



陆嘉恒

中国人民大学副教授，教育部新世纪人才、北京科技新星，新加坡国立大学博士，2006-2008年在加利福尼亚大学尔湾分校进行博士后研究。主要研究领域为数据库和信息检索技术，主编教材《分布式系统与云计算》。



许翔

中国人民大学硕士研究生，主要研究领域包括MapReduce编程框架和Hadoop云计算平台。

# 大规模图计算模型——Pregel

■ 文 / 肖康

图是计算机科学中的经典问题，在现实也有很多应用与图相关，例如Web链接组成的图、SNS中人与人之间的各种关系图、电子地图中的路径搜索等。当图的规模很大时就需要一种有效的方式来处理与图相关的计算，有资料显示在Google大规模数据处理80%是用MapReduce，而其余20%是用另一个面向图计算的模型——Pregel。Google 2010年发表在SIGMOD的论文“Pregel: a system for large-scale graph processing”对这一模型进行了揭秘，论文中提到已经有大量的生产应用通过Pregel开发运行，而且很少有不能在Pregel上实现的实际图计算应用。

目前针对图计算的解决方案中，单机图算法库比较成熟但扩展性不好，很难处理Web规模的图计算；并行图算法库又缺少容错，在大规模数据处理中出错又是很正常的；基于MapReduce的多轮迭代又不够高效，因为需要反复存储和读取迭代的结果；MPI实现并行计算比较灵活，但需要应用考虑通信、同步、容错等，因此编程比较复杂。

与以上计算模型相比，Pregel具有以下几个特点和优势：第一，原生的图计算API，简单易用且表达力强；第二，高效；第三，很好的容错和可靠性；第四，可扩展性好。

## 模型

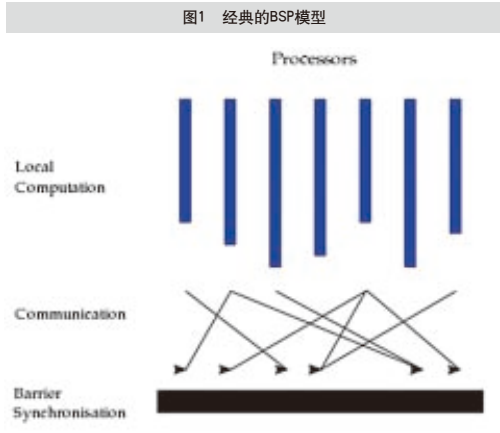
Pregel计算模型可以概括为BSP（Bulk Synchronization Parallel）+ 节点为中心的图计算API + 容错。

如图1所示，经典的BSP模型计算由多次迭代组成，每次迭代称为一个superstep，每个superstep分为以下3个步骤。

**计算：**各个进程独立同步进行，且每个superstep执行同样的逻辑。

**通信：**各个进程之间进行通信，通信的数据在下一个superstep作为输入使用。

**同步：**所有进程进行同步，然后从1开始下一个superstep。



Pregel提供了如图2所示的以节点为中心的API，它具有以下几个特点。

■ 以节点为中心，所有的边是源节点的附属品。一个节点的数据包括三项：节点的值、它的出边的值、下一个superstep要处理的消息，在API中分别由Vertex Value、Edge Value、Message Value类型表示。

■ Compute()方法是计算的核心。可以进行的操作包括：读取上一个superstep发送到该节点的消息，向其他节点发送消息，读取和修改节点的值，修改出边的值，修改图的拓扑。

■ 所有节点初始为Active状态，节点在Compute()中调用VoteToHalt()进入Inactive状态，当有消息发送到Inactive状态的节点时，它又被激活到Active状态；如果所有节点都进入Inactive状态且没有任何消息，则整个计算结束。

图2 Pregel提供的以节点为中心的API

```
template <typename VertexValue,
          typename EdgeValue,
          typename MessageValue>
class Vertex {
public:
    virtual void Compute(MessageIterator* msgs) = 0;

    const string& vertex_id() const;
    int64 superstep() const;

    const VertexValue& GetValue();
    VertexValue* MutableValue();
    OutEdgeIterator GetOutEdgeIterator();

    void SendMessageTo(const string& dest_vertex,
                       const MessageValue& message);
    void VoteToHalt();
};
```

与BSP对应，Pregel计算模型的superstep也分为下面三个步骤。

**计算：**对于图的每个节点Pregel执行应用提供的Compute()方法，每个节点独立并行地进行计算。

**通信：**通信只能通过收发消息进行。节点之间通信是和计算并发进行的，计算过程中可以向图中的其他节点发送消息，这些消息的收发由Pregel自动完成而不需要应用关心，当前superstep发送的消息作为下一个superstep的输入消息，而在当前superstep不可见。Pregel不保证发送到一个节点的消息的顺序，但是保证一定会发送到且不会重复。节点一般向其相邻节点发送消息，也可以向其他任何节点发送消息。

**同步：**当一个superstep的所有节点计算结束且所有消息都已经收发完成，就可以开始下一个superstep。

Pregel提供了自动容错的机制，当部分节点的计算由于各种原因如机器宕机、网络异常等失败后，会通过自动重算部分superstep进行恢复，整个计算时间可能会拉长但还是会完成，并不会因为系统异常而失败。

如图3所示，一个典型的Pregel计算由输入、supersteps、输出组成。输入是要处理的图，会按照节点进行划分，输出是最终各个节点的值，多数情况下也是一个和输入类似的图，supersteps就是上面描述的计算过程。

## 实现

### 总体架构

Pregel的实现从架构上是master/slave的结构，如图4所示。输入的所有节点分割成一个partition，master负责将每个partition分配给各个worker，协调各个worker执行superstep，进行全局信息汇总和状态显示；worker负责保存节点的状态，执行superstep。

图4 Pregel的架构

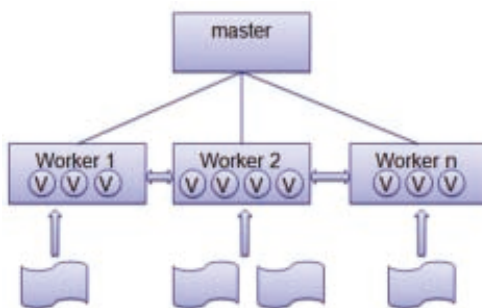
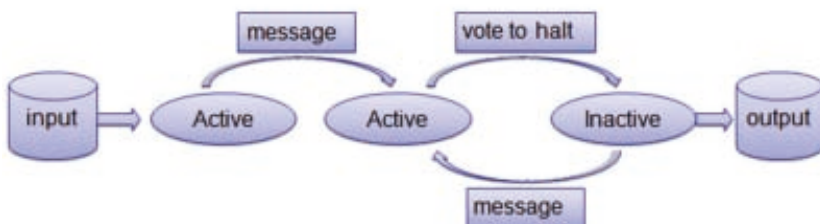


图3 一个典型的Pregel计算的组成





### master/worker

master负责将partition分配给各个worker，这里允许应用指定分配的策略，因为不同的分配方式可能对计算性能产生影响。分配的单位是partition而不是节点，因此单master就可以处理很大的图。一个worker可能会分配到多个partition，这样并发度更高且有利于worker之间负载均衡。

master对superstep进行同步，一个superstep的所有节点计算和消息收发都完成后，便通知worker开始下一个superstep。在一个superstep中，worker循环执行它负责的所有节点的Compute()方法，每个线程处理一个partition。

---

Pregel具有以下几个特点和优势：第一，原生的图计算API，简单易用且表达力强；第二，高效；第三，很好的容错和可靠性；第四，可扩展性好。

---

worker在内存中维护每个节点的数据，包括节点的值、出边、接收的消息、Active/Inactive状态，计算的进程一直运行而不是每个superstep启动新的进程，这点和基于MapReduce的多轮迭代有很大区别，减少了每次迭代启动进程和加载、保存数据的开销。

对于每个节点接收的消息，worker维护两个队列，一个是上一个superstep收到的消息，一个是当前superstep正在接收的消息。这样每个节点在计算过程中使用上个superstep消息的同时，可以异步接收当前superstep发送给它的消息，使计算和通信并行起来。Compute()中发出的消息，如果是发往本机节点的就直接放入其消息队列，如果是发往其他机器worker则先缓存起来，积累到一定量才异步地发送出去，以提高网络传输的效率。

### Combiner和Aggregator

节点规模比较大时，发送的消息量可能比较多，Pregel提供了Combiner机制减少消息量，应用可以提供Combiner，消息发送端

和接收端可合并一些消息，以减少发送消息需要的网络带宽和存储消息的内存资源。这一点和MapReduce中的Combiner比较类似。由于Combiner需要满足交换律和结合律（比如累加计数），所以无法提供默认实现，只能由应用提供。

Pregel还提供了一个和MapReduce Counters类似的Aggregators机制，用来汇聚全局信息。Worker维护每个节点的Aggregators，本地汇聚后上报给master进行全局汇聚，最后把全局汇聚的值发给各个worker。Aggregators和Counters有两点不同：一方面Aggregators运行提供汇聚函数（如取最大值），而Counters只是累加；另一方面全局汇聚值还会发给worker在下一个superstep使用。

### 容错机制

Pregel通过checkpoint来实现容错：master定期向slave发ping，worker收到ping后向master回应；worker如果长时间未收到ping就自动退出；master如果长时间未收到回应就认为worker出错。superstep开始之前，worker把节点数据（包括节点、出边，收到的消息）保存持久化存储如GFS中，如果master检测到worker出错，会重新将所有partition分配给正常的worker，让每个worker从持久化存储中加载对应partition最近一次保存的节点状态，然后从这个superstep开始执行。

即使上面的容错过程中没有出错的worker节点，也要从上次checkpoint的superstep重新执行，因此Pregel正在开发一种更高效的方式（Confined Recovery）只重新执行出错的supersteps。这种方式需要worker记录上次checkpoint开始所有发出的消息，当某个worker出错时，在其他worker上加载对应节点的checkpoint，其他正常的worker把记录的消息重新发送给这些节点，这些节点就能从上次checkpoint恢复到worker出错时的superstep，而其他worker上的节点不需要恢复。这种方式的优点是只用恢复出错的worker，可以减少重算所有节点带来的浪费，因此恢复的时间也能缩短，而缺点是需要记录更多的数据而且过程相

对复杂一些。

## 应用

这里以论文中单源最短路径的实现说明Pregel的应用。求单源最短路径是找出图中某个点到其他所有点的最短距离，在Pregel中实现如下（图5）。

图5 单源最短路径的实现

```
class ShortestPathVertex
: public Vertex<int, int, int> {
void Compute(MessageIterator* msgs) {
int mindist = IsSource(vertex_id()) ? 0 : INF;
for (; !msgs->Done(); msgs->Next())
mindist = min(mindist, msgs->Value());
if (mindist < GetValue()) {
*MutableValue() = mindist;
OutEdgeIterator iter = GetOutEdgeIterator();
for (; !iter.Done(); iter.Next())
SendMessageTo(iter.Target(),
mindist + iter.GetValue());
}
VoteToHalt();
}
};
```

■ 点、边、消息的类型都是整数，边的值是两个相邻点的距离，初始值由输入图决定。点的值是源点到该点的最短距离，源点初始值为0，其他点为无穷大。

■ 每次superstep中，节点收到来自相邻节点的消息，消息的值是源节点到相邻节点的最短距离。如果消息的最小值比该节点本身的值小，说明源节点通过某个相邻节点到该节点的距离比当前最短距离还小，就将自己的值替换为最小值，且将自己的值加上出边值发送给出边的目的节点。

■ Superstep一直到没有节点向外发消息，也就是每个节点的最短距离就是当前节点值时结束。

根据论文中的测试数据，Pregel的扩展性还是相当不错的，worker数不变时对于SSSP图的节点数增加计算时间随之线性增长，而图的节点数不变时worker数增加16倍使性能提高10倍。

Pregel在Google的使用过程中不断得到发展，比如加入Aggregators、Web展示页面、单元测试和单机调试工具等，这些经验也是设计其他计算模型和系统时值得借鉴的，因为易用性的改进对于一个模型和系统的广泛使用也是

很关键的。Pregel目前处理上亿节点规模的图已经不是问题，为了能处理更大规模的图，还会有一些优化和改进：内存不够用时worker利用本地磁盘存储一些数据；动态重新分配partition到不同的worker以获得更优的性能；简化BSP的同步模型，使得那些运行得快的worker不用等待少数慢的worker等。

## 总结

Pregel的优势在于简单、高效、容错、可扩展，它的计算模型和MapReduce一样比较简单、容易理解，作为一个提供了自动通信、容错等机制的分布式计算模型，它的API比较友好，使应用专注于单个节点的本地操作而不是通信、容错等分布式相关的问题。虽然没有MPI灵活，但它提供的自动消息收发、容错等机制对应用来说很有价值。在扩展性方面，与相关图计算的库和框架相比能扩展到更多的机器。

当然，Pregel不如MapReduce应用广泛，可能也有一些原因，比如：模型和API专注于多轮迭代的图计算，有些计算可能不方便向这个模型映射；对于稠密图和需要大量通信的计算，还是不太高效等。

不管怎样，作为一个新的计算模型Pregel是值得研究和使用的，目前已经有一些相关的开源实现，例如Apache基于Hadoop的BSP模型Hama，Github上多个Pregel项目，RAVEL基于Hadoop的Pregel实现GoldenOrb。虽然都还没有像Hadoop这样成熟的产品，还是值得关注和贡献的。P



肖康

百度基础架构部高级工程师，百度分布式计算方向核心开发人员，百度下一代计算系统模块负责人。曾受邀在Hadoop in China 2010会议上进行技术讲座。

# 推荐系统在视频网站中的应用

■ 文 / 郑华

随着互联网特别是社会化网络的快速发展，我们正处于信息过载的时代。用户面对过量的信息很难找到自己真正感兴趣的内容，而内容提供商也很难把优质的内容准确推送给感兴趣的用户。推荐系统被认为是解决这些问题的有效方法，它对用户的历史行为进行挖掘，对用户兴趣进行建模，并对用户未来的行为进行预测，从而建立了用户和内容的关系。

视频网站同样也面临着信息过载的问题，比如YouTube目前有数十亿视频条目，而且每分钟都有近1500分钟时长的视频被上传，而专业视频网站Hulu也有近20万的高质量视频，用户也很难从中找到自己真正感兴趣的内容；同时提高用户的观看和停留时间也非常重要，所以推荐系统对于视频网站来说是有必要的。

## 视频网站中的推荐系统

目前视频网站主要分为两种，一种是以用户产生内容为主的网站（UGC网站），比如国外的YouTube和国内的优酷、土豆等；另一种是专业视频内容为主的网站，比如国外的Hulu、Netflix和国内的奇艺、新浪大片、搜狐视频等。为了获得广告商和资本市场的青睐，国内UGC视频网站比如优酷和土豆也纷纷转型为两者兼备的模式。这两种视频网站的内容和用户行为都是不一样的，从而会导致相应的推荐系统的设计也会有一定差别。表1简单比较了两种网站的一些差别。

相比较而言，UGC网站的视频数量多，内

容比较丰富，但是质量良莠不齐，以短视频居多，而且没很好的内容数据。一般的推荐是基于单个视频的，而且会对视频质量做一定过滤（比如去重和限制最少观看次数等）；UGC内容生命周期比较短，所以推荐系统的设计比较强调时效性，把最新的视频推荐给用户，保持推荐的新鲜性；同时UGC网站的内容比较多样化并且用户一般没有很强的目的性，所以推荐要尽量多样化并且和用户最近的行为相关。

对于专业视频网站，内容一般有很好的结构化内容数据，基本上都是按照电视剧或者电影来统一组织视频，所以推荐一般都是基于剧集而不是单个视频；剧集按照现在是否在播又分为首播剧（On-air show）和重播剧（Library show），相比较而言首播剧一般比较热门，用户获知的渠道很多并对其有明确的追看（Catch-up）需求，而过往剧集更适合推荐；视频的长度一般比较长而且剧集中有许多视频，用户的接受成本相对比较高，推荐的时机也是专业视频网站需要考虑的问题，比如周末或者节假日用户比较空闲而且一般没有首播剧，适合推荐一些较长的连续剧等。

当然除了这些不同点外，一般认为视频网站的推荐系统设计还应遵循一些基本的原则，比如系统能给出合理的推荐赢得用户的信任；系统能对用户的行为作出即时的反应；推荐的逻辑对用户透明；在适当时机鼓励用户积极参与和反馈；推荐的结果要提供足够的信息等。

从推荐产品形态上来看，目前在视频网站中使用的推荐有以下几类：相关推荐、个性化推荐以及个性化电视频道（Leanback）等。

相关推荐就是把用户正在观看或者浏览的视频的相关视频推荐给用户，也就是“喜欢这个视频的用户还喜欢”，比如图1是Netflix对经典《两杆大烟枪》的相关推荐界面，推荐出来一些盖·里奇和其他导演的黑色幽默作品。结果

表1 UGC和专业视频网站对比		
	UGC 视频网站	专业视频网站
视频数目	一般大于1亿	一般小于100万
视频长度	短视频为主，观看行为容易引入噪音	长视频为主
内容数据	很差，很多只有标题	结构化数据
生命周期	很短	一般比较长
视频质量	良莠不齐，大量重复内容	很好
视频多样性	种类比较丰富	内容多样性不够



展示出视频的重要信息（标题、缩略图、平均打分、类型、年代、时长、简介、导演、演员等），并且提供打分或者不感兴趣的选项来收集用户反馈。

个性化推荐是根据用户所有的历史行为推断出用户的兴趣，并以此推荐用户最可能感兴趣的视频列表。与相关推荐相比，个性化推荐综合使用了用户的所有历史行为，包括打分、观看、订阅、搜索、标注标签、分享和评论等，而不仅仅是当前的单个浏览或观看行为，所以更准确反映出用户的兴趣。从历史行为中反映的用户兴趣也可能会多样化，所以一般个性化推荐展示结果会按照相对独立的兴趣点（比如类型等）进行聚合。图2是Hulu的个性化推荐的展示形式：我们按照推荐视频的类型对结果进行聚合，用户也可以切换自己感兴趣的类型；结果有很详细的视频内容信息以及最有用的用户评论信息；推荐结果有解释；用户可以直接对推荐结果进行“已经看过”或者“感兴趣与否”的反馈，如果感兴趣还可以进一步收藏推荐结果或者直接进入观看页面等。

在线视频的一个重要战场就是用户的客厅电视，这是最有可能采取收费模式的。目前包括Netflix、Hulu Plus等服务都提供720P的高清视频可以通过游戏平台、机顶设备、蓝光播放器或者高清电视直接播放在大尺寸的客厅电视上。Google也在去年推出了Google TV，试图进入这一领域。为了保持和现有电视相似的操作，Leanback就应运而生了。图3就是YouTube推出的Leanback服务的界面。用户可以创建自己的频道，或者推荐系统根据用户的历史行为自动创建符合用户兴趣的视频节目频道。用户可以不间断地观看频道里面的视频，播放的过程中系统不断收集用户对视频的反馈（喜欢、不喜欢、跳过、看完等）实时调整推荐列表，让用户看到越来越满意的结果。

业界一般认为用户界面设计在整个推荐系统中起决定性的作用。传统的推荐结果展示都是基于文本和图片形式的，信息量大但不够形象生动。Hulu在推荐的UI上做了一个大胆的尝试，首次使用视频和声音的形式展示推荐的结果，效果类似于有个性化消息的电视预告片，

图1 Netflix相关视频推荐



图2 Hulu的个性化推荐



图3 YouTube的个性化电视频道

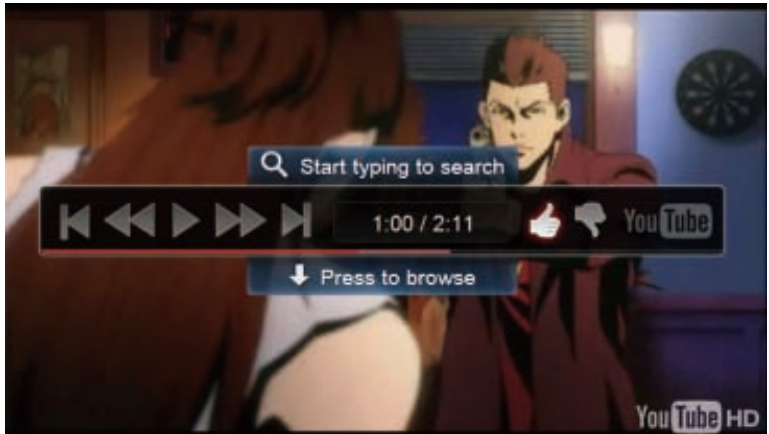


图4 Hulu的视频展示形式的推荐



如图4所示。在实际使用中取得了较好的效果，平均意义上用户对同一个节目感兴趣的概率是传统方式的三倍。



## 推荐系统的方法

一般认为推荐系统的方法可以按照数据和模型两个维度进行分类。从使用的数据上来看，推荐系统可以分为协同过滤系统、内容过滤系统和社会化过滤系统等；从使用的模型来看可分为基于邻域模型、矩阵分解模型和图模型等。

协同过滤（Collaborative Filtering）是推荐系统中最著名的方法，它主要通过用户的历史行为分析出用户的兴趣并给用户做出推荐。协同过滤有很多算法，比较常见的有邻域算法（UserCF和ItemCF等）、矩阵分解算法（或Latent Factor Model，如RSVD和SVD++等）和图算法等。目前视频网站中比较常用的协同过滤算法是ItemCF，它的基本假设就是用户会喜欢跟自己之前喜欢视频比较类似的视频。因此在给这个用户做推荐的时候，需要先从用户的历史行为中得到他喜欢的视频列表，然后从剩下的视频中找到和用户之前喜欢列表最相似的视频推荐给他。可见这个方法最核心的就是怎样合理计算两个视频的相似度，比较常用的有余弦相似度或者皮尔逊相关系数等，实际使用的时候需要根据具体情况进行修正。一般认为ItemCF算法比较简单、容易扩展，准确度比较高，能实时更新而且可以解释，可以处理显式（打分或者感兴趣）或者隐式反馈（其他如观看等行为），所以在实际的视频推荐系统中如Netflix、Hulu和YouTube都使用了它。协同过滤方法的一个重要缺点是不能冷启动，也就是对于新加入的视频或者用户都不能做出推荐，一般需要混合其他推荐方法（比如内容过滤等）来处理这个问题。

内容过滤的基本思想是给用户推荐和他们之前喜欢的视频在内容上相似的其他视频。比如用户喜欢看《两杆大烟枪》，那么内容过滤系统就会推荐盖·里奇的其他类似内容作品如《偷拐抢骗》等；如果用户喜欢《火影忍者》，系统就会推荐《火影忍者疾风传》或者其他热血类的日本动画。因此内容过滤的核心是怎样计算两个视频之间的内容相似度。一般情况下，计算视频的内容相似度是从视频内容

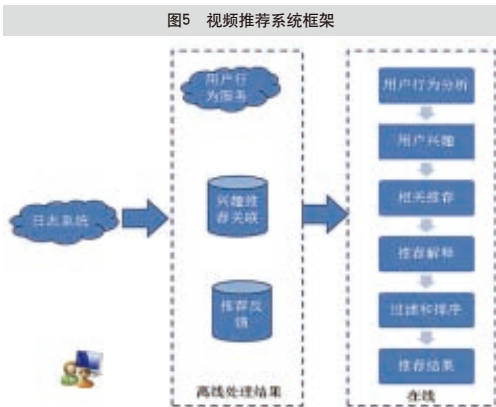
（比如标题、类型、地区、出品公司、年代、导演、演员、剧情简介、用户标签、评论等）中抽取出关键词，然后确定这些关键词的权重，这样得到了这个视频的向量模型，再计算两个视频向量模型的相似度。随着专家标注系统Pandora在音乐推荐领域的大获成功，目前在视频领域也出现了类似的网站比如Jinni，它定义了描述电影基因的900多个标签（类型、剧情、类别、年代、地点、心情、适合的观影人群、好评、风格、态度、画面等），然后电影专家会给每部电影标注这些标签，从而可以得到每部电影的专家标注的向量空间并以此做出推荐。由于专家标注的工作量非常大又没有公认的收益，目前在实际中并没有大规模使用，一般视频网站还是使用比较传统的方法，结合视频的内容和用户的标签进行内容过滤。

社会化过滤的思想是用户的喜好可能会受他在社会网络中的好友影响。随着SNS网络的兴起，社会化网络的推荐越来越受到关注，比如视频搜索网站Clicker就利用Facebook的好友关系做出推荐；当然利用Facebook的另外一个好处是视频网站可以得到用户的更多信息，特别是一些站外的Like信息，这些也会帮助改善推荐质量。

一般的视频网站的推荐系统都不是单一的，而是混合了这些过滤方法，目前比较常见是混合ItemCF和内容过滤等。

## 视频推荐系统的框架

视频推荐系统主要工作是从用户的历史行为中分析出用户的兴趣然后找出符合其兴趣的视频展示给用户。因此一个完整的推荐系统，至少包括日志系统、推荐引擎和展示界面设计等部分。展示界面设计在前面我们已经谈过，这里就不再赘述。日志系统主要收集用户的行为和对推荐系统的反馈。推荐引擎也分离线和在线两部分：离线系统主要负责生成视频相关矩阵，存储在数据库中，供在线系统实时查询和调用；在线系统负责实时响应用户的请求，在线提取和分析用户行为并生成最终推荐结果。一个典型的视频推荐系统的框图如图5所示。



目前一般视频网站的流量和用户规模都比较庞大，后台的日志系统的处理基本上都会使用云计算平台进行处理，比如Hulu就搭建了自己的Hadoop集群来处理用户的行为日志文件。该集群由50个节点组成，每个节点有6TB的空间，每天处理大约500GB数据；系统支持定期或者临时从指定时间段的日志中提取符合模式的数据，最终按照一定的维度聚合发布到分布式文件系统HDFS、HBase或其他SQL数据库供后续使用。日志系统收集了表示用户兴趣的行为比如观看、打分、搜索、标注标签、订阅、分享、评论以及用户对推荐系统的反馈行为（点击、感兴趣、已经看过、收藏）等。

推荐引擎离线部分利用收集到的用户行为日志：计算出一系列的关联矩阵（比如视频之间的两两相似度、电影主题和视频的之间关联等）；计算全局或者某些群体用户的对推荐系统反馈（比如用户行为的权重、推荐算法的权重等）。对于其中数据量比较大的任务，可以直接在Hadoop系统上并行处理，具体实现可以参考开源项目Apache Mahout。

“用户行为服务”模块实时返回用户的重要行为（打分、感兴趣等），由于用户数很大，一般需要使用数据库拆分（Database Sharding）加前端Memcached的方式来解决扩展性问题，数据库可以是传统的MySQL或者是NoSQL数据库（如MongoDB等）。

在线部分需要根据用户行为分析用户的兴趣特征并结合离线训练结果进行推荐、解释、过滤和排序等，然后给出最终的推荐列表。这块依赖的数据相对比较单一，出于可扩展性和

灵活性考虑，可以把服务部署到云计算平台（如AWS、Google App Engine或Windows Azure等），实际上Netflix就把包括推荐系统的大多数服务部署到AWS。

## 推荐系统的作用

推荐系统一般使用以下指标对推荐系统进行评测：准确度、覆盖率、多样性和新颖性等。在实际视频网站推荐系统中，通常更关注推荐系统带来的在线直接收益，比如点击率、转换率以及用户实际停留和观看时间的增长等。那么推荐系统对视频网站究竟有没有作用呢？答案是肯定的。YouTube公布说首页60%的点击来自于推荐，并且比较了个性化推荐和最热门视频的点击率，结果前者的点击率是后者的207%；Hulu也有类似的实验结果。而Netflix整个网站的设计完全是围绕推荐展开的，这也正说明了推荐在视频网站中的作用。

## 结语

视频网站充斥着大量用户可能不感兴趣的内容，同时内容提供商希望把自己的优质内容推销给感兴趣的用户，而视频网站也需要提高用户的黏度和观看时间，这些都决定了视频网站需要推荐系统。目前国外大多数视频网站都有自己的推荐系统，而国内的同行也在加大投入研发和部署推荐系统，这对推荐系统领域研究人员是极大的鼓舞。本文简单介绍视频网站中推荐系统的产品、方法和简单框架，希望能对感兴趣的人有所启发。P



郑华

2007年毕业于清华大学电子工程系并获得硕士学位，之后加入hulu工作至今，是hulu最早的员工之一。目前领导hulu的推荐团队，该团队主要负责视频推荐、广告精确投放、数据分析以及支付系统等项目。

# Amazon EC2: 系统测试的云中基石

■ 文 / 刘星 孙靓

云计算的出现，带来了相关技术的长足发展，同时也带来了多方位的挑战。这种状况，在云计算从概念到实现的转变之后，尤为明显。无论是业界的领导者，如Google、Yahoo，还是那些刚露出尖尖角的后来者，都将面临着一些共同的难题和挑战。在这些难题与挑战之中，如何在云中合理搭配计算资源则一直是让业者纠结的话题。

## 开发者的云与测试者的云

开发阶段的云平台应用产品，大多数情况下并不需要太多的计算资源，例如为客户开发一个基于云平台的并行存储系统，在开发阶段，往往只需3~4台实体机便可以搭建出足以支撑程序员进行全部功能开发的环境。在开发阶段完成后，白盒的单元测试和黑盒的功能测试也都可以在这个环境里很好地进行，且当产品在真实的云平台上运行之后，单元和功能测试的结果也都拥有足够的说服力。但是，我们都知道，传统的企业级应用产品，在上线之前，系统测试是不可或缺的。云平台上的应用产品会有例外吗？财力雄厚的诸如Google这样的国际性公司，在产品上线前，必然会进行一系列的性能测试、压力测试、稳定性测试以及安全性测试，这样一来，3~4台实体机组成的开发环境就显得捉襟见肘了。他们必然会通过自家的云平台模拟实际运行环境，以供系统测试执行。

显然，无论是传统软件产品还是云平台上的应用产品，系统测试都是必需的一环。云平台产品的特点，决定了其在集群中的优异表现来自于可智能化地伸缩和迁移计算资源。专业的系统测试可以获知产品的真实能力，同时

带来对计算资源的伸缩性预估。然而对于不如Google那样财大气粗的软件公司来说，大量采购计算资源以实现对产品的系统测试，是不够现实的。原因在于以下几点。

■ 产品的吞吐需求可能是无法从客户那儿预知的，这意味着，软件开发者无法获知产品将处于何种规模的云朵中，甚至这片云彩的大小是否足以容纳产品都将是未知数。

■ 程序员需要知道产品在云平台上的瓶颈所在，而客户则想了解如何使用最少的计算资源就可以满足实际情况的需求，这需要数据来支持程序员和客户的想法。平衡点的权衡是一个重要的课题，而程序员和客户由于视角的不同，必将在测试规模上产生分歧，无法确定因此产生的计算资源需求量。

■ 过度采购必然会产生浪费，程序员当然有理由相信客户已经拥有了足够的计算资源，这样产品可以直接在实际应用环境中进行各种系统测试。然而，客户也同样有理由相信企业在推出自己产品的同时，也可以告知真实的计算资源需求——两者都不愿看到由于过度采购而产生的不必要浪费。

■ 维护一套云平台需要投入大量的人力资源，硬件的损坏对于海量的计算资源来说几乎必然在每时每刻都会发生。聘请大量有经验的人员来维护这些可能大部分时间由于无测试任务而处于闲置状态的计算资源，显然也是一种浪费。除此之外，网络的搭建、环境安全上的顾虑都将大大增加维护的成本。

■ 这样一套用于系统测试的模拟环境，对于QA来说，是需要投入大量精力用于搭配和部署各种组件的。这无疑将带来额外的消耗，同时也将使QA无法全神贯注地投入到产品的测试中。

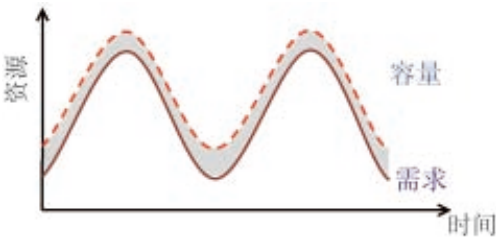


## 灵感与曙光

好在大多数IT从业者都充满激情，面对困难总会迸发出超乎寻常的灵感火花，以接受挑战并解决难题。上述的困难，对于这些才华横溢的行业开拓者来说，似乎也总该有合理的解决办法。曙光终于在本世纪的第六个年头开始出现，由Amazon带来的EC2（Amazon Elastic Compute Cloud）服务使得解决以上困难变得更加容易且低成本。

Amazon是一家伟大的公司，它不仅仅在卖书，还在改变历史。EC2服务无疑正是Amazon改变历史的一个强有力证据，它告诉我们Amazon还能够贩卖富余的空闲计算资源。在Amazon EC2服务出现前，计算资源只能通过购买实体硬件获得。但客观的现实是，我们所需的计算资源经常会因实际需求而发生变化。例如，对于C2C网站来说，某些特殊的节日，对于其主站点的访问量会达到平时的数倍。但如果根据最高访问量来购买实体计算资源，那有可能在大多数的空闲时段将会处于闲置状态。这还没有算上硬件的折旧损耗，如此一来将会产生严重的浪费，如图1所示。

图1 需求总是呈现出波浪形，采购计算资源可以满足最大需求，但多余的资源在波谷处又总是被浪费



有没有什么方案能够让我们根据实际需求来启用合适的计算资源呢？Amazon给出了答案。无论是Core、Memory还是Disk，对于Amazon来说都仿佛是可以被租赁的汽车，当你恰好需要的时候，你将以极低的成本（相对于购买一辆汽车来说）获得它们的使用权。在你使用完毕后，完全不必考虑这辆汽车是否有损坏，只需要归还这些汽车就行了。甚至Amazon还顺带帮你完成了租赁资源的升级和保养，使用者永远不用担心租到的汽车会发生故障（也

许还是有一些微不足道的故障概率），或者外观不够新潮，马力不够强劲。

闲话少说，让我们从一个实例来看看Amazon EC2服务改变的历史。

## 建立在云中基石上

这个实例来自于趋势科技中国研发中心的一群充满热情的工程师们，他们在2009年接到了一个分布式存储数据库的开发任务，很快他们在研究了各种可行方案之后，制定了技术路线，确定将基于Hadoop/HBase这样的并行计算框架来开发这个产品。这个产品被正式命名为GMS（Global Meta Store）。

GMS在HBase的基础上补充了一些更为有用的功能。例如，HBase作为典型的Key-Value型数据库，其本身只提供了单索引的快速检索能力；而GMS为其扩展了多索引实时检索的支持。再例如，HBase本身是不支持SQL语句的，使用者为了查询自己需要的数据，标准做法是编写一个包含了查询语句的jar包提交到Hadoop的MapReduce计算框架下进行查询；而GMS则支持部分SQL语句的直接提交和执行，这将极大地提高产品的用户体验。此外，HBase本身不提供对BigTable的备份还原功能；而GMS也对这个极为常见的需求进行了开发，使得数据在HBase中的可靠性大为提高。

然而，在GMS产品开发阶段完成之后，他们的QA同样也遇到了缺少云平台进行系统测试的难题。他们无法回答，在真实的云平台上，支持多索引的GMS会比HBase的数据插入性能下降了多少；他们也无法回答当数据量达到何种规模的时候，客户需要为他们的云增加计算资源，更无法指导客户该如何增减计算资源。甚至由于不良配置的Hadoop/HBase可能造成性能瓶颈或者稳定性不佳，也是他们所无法预知的。而等到产品发布之后再进行这些系统测试，又将会影响其他依赖于GMS的产品，使得这些产品不得不重新进行设计，更严重的情况下，甚至会影响云平台下全套产品的基本架构。同时，采购足量的计算资源用于这个产品的系统测试也是不现实的，一方面，从成本角度考虑，采购带来的花费太大；

另一方面，从采购到搭建到完成测试，所需的时间周期太长，会影响到其他依赖于GMS的模块的开发进度。

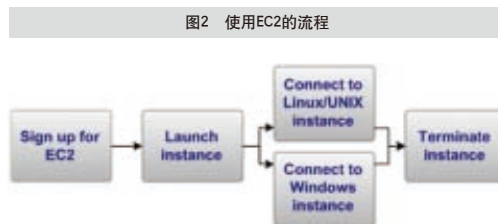
对于GMS这样的项目来说，系统测试——尤其是性能测试，又是极为必要的。无论是依赖于GMS的产品，还是云平台的维护者们此时都需要了解GMS的具体能力，他们需要看到可靠的数据来说明GMS的真实性能。然而对于GMS开发团队的QA来说，除了上文的两方面考虑之外，凑集足够规模的计算资源来模拟现实运作环境还有两点困难：其一是因为我们需要不止一朵云来模拟不同的运作环境，其二是因为即使是Hadoop/HBase的最低推荐运行环境所要求的每个节点的硬件配置也是相当高的。这个时候，GMS团队想到了利用Amazon EC2提供的计算资源租赁服务来实现系统测试。

## 恰当的解决方案

GMS产品，包括Hadoop和HBase，都有一个共同的特点，而这个特点也往往是大多数云平台应用的共同点，那便是这些模块的配置在所有节点上都是完全一致的。这意味着，只要在一个节点上配置完毕，便可以将这个节点直接克隆到其他所有节点上。这当然也意味着，虚拟机的镜像文件在这里将大有作为。

Amazon EC2为使用者提供了类似于虚拟机镜像的AMI（Amazon Machine Image），这样就可以在EC2上申请一个已安装了CentOS系统（也可以是其他系统，如Ubuntu、Fedora或Windows）的Instance，在配置好我们需要的组件后将其保存为AMI，然后从这个AMI启动所有的Instance，便可以轻易地搭建出产品运行所需的集群。计算资源的伸缩在EC2上也是相当容易的一件事情，可以简单地通过Amazon提供的RESTful Interface来启停Instance，也可以通过EC2提供的Elastic Load Balancer服务来根据集群整体负载状况自动启停Instance。在EC2上运行的Instance一旦被使用者销毁之后，其上发生的所有改变都将随同Instance一起消失。这看起来不像是一个好消息，但实际上对于测试来说却是省去了集群初始化的麻烦。当然EC2也提供了永久保存 Instance的办法（前

提是用户持续为占用的空间付费），称之为EBS（Elastic Block Store），如图2所示。



Amazon EC2提供的强大功能远不止上述这几项，由于其在五个地方（US Virginia、US California、EU Ireland、APAC Singapore、APAC Tokyo）拥有实体物理机，因此我们运行的Instance也可以任意分布在这些地方。从大的方面来说，分散节点到物理世界的不同位置，有利于在发生自然灾害时不至于丢失数据；从小的方面来说，也可以方便用户模拟真实的网络分布环境。同时，各地区申请运行Instance的花费也会存在一些差异，这些信息都可以在Amazon EC2的首页上查询到，最便宜的租赁地点当属美国的Virginia。为了更节约花费，Amazon还推出了竞价机制，在申请Instance的时候，如果EC2的计算资源比较富余，那么通过竞价，使用者可以低于普通方式的费用来获得Instance的使用权；相反，在计算资源紧张的时候，竞价方式可能会提高到接近于普通方式的花费，但最多也不会超过普通方式。竞价机制的缺点在于，申请者有可能一次性无法申请到足够数量的Instance，或者申请到的Instance可能会在不同的物理位置，这将带来一些网络I/O的花费，但是比起普通方式申请Instance来说，竞价机制节省下来的成本还是非常令人心动的。

前面提到，在EC2上运行的Instance一旦被使用者销毁之后，其上发生的所有改变都将随同Instance一起消失。这里的消失，不仅包括数据的消失，也包括IP地址的消失。因此，在某些情况下，我们有必要申请Elastic IP用于公网的永久性服务。同时为了这些服务的高可靠性，我们可以使用Elastic Load Balancer服务，当负载加重的时候，EC2会按照我们设定的规则自动增加节点数量；反之，EC2将会销毁节点以节省用户的开销。

简单的流程

GMS的性能测试是完全使用EC2提供的云平台完成的。QA们制作了一个自动化脚本，通过Amazon EC2的RESTful Interface来实现云的搭建。这个脚本完成的功能和流程大致如图3。

- 1. 从配好的AMI启动足够数量的Instance。
- 2. 自动获取所有Instance的内网IP地址和Hostname。
- 3. 根据网络信息自动分配各Instance担负的角色，部署Hadoop和HBase到各节点，将其启动并检查进程运行状态。
- 4. 部署GMS产品和性能测试客户端到集群中。
- 5. 执行性能测试，并收集测试结果。
- 6. 通过邮件方式传回测试结果，随后销毁Instance。

取得的成果

通过这个脚本，GMS团队实现了模拟百数量级集群的性能测试和稳定性测试，而所需的花费仅是一台相当于Instance配置的实体物理机价格的不到二分之一，如下表所示。

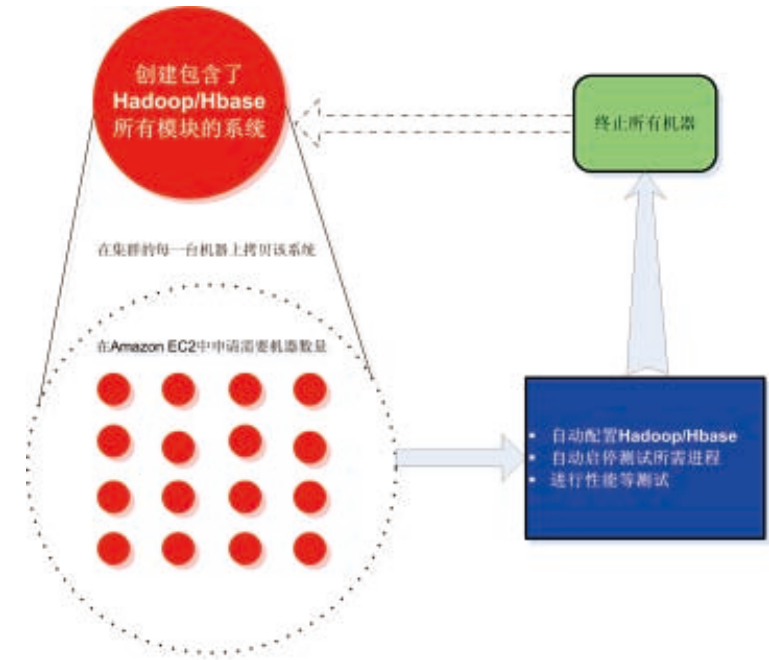
硬件配置	平时购机费用	Amazon租赁费用
英特尔® 至强® CPU E5430 @ 2.66GHZ	\$2000/台	\$0.34/小时
内存 8 GB		
硬盘空间 > 400 GB		

使用恰当的方法解决实际问题，这正是GMS团队面对困难时迸发出的灵感火花。在这个实例中，QA们不用去考虑实体机的安装部署，简单的几行命令便可以置身于云中，甚至连安全方面的顾虑也可以避免了，因为EC2为其中的私云内置了防火墙，使用者可以通过AWS（Amazon Web Services）迅速配置允许外界访问私云的端口。

结束语

使用EC2不仅节约了花费，更令GMS团队开拓了视野和思路，他们将这种测试方法推广到了趋势科技中国研发中心的其他开发团队中，目前已有部分产品的系统测试是通过使用

图3 在EC2中进行性能测试的流程图



EC2平台来实现了，将来这样的应用场景会在趋势科技的开发团队中更加常见。

展望未来，我们不难发现，类似于Amazon EC2这样的计算资源提供商会越来越多（例如微软的Azure和天云的TCloud），系统测试对于这样的云中基石来说，不过是牛刀小试，如今更有像Twitter这样的著名公司将自己的全部架构建立在EC2中。可以想象，不久的将来，如何合理利用这些计算资源来完成我们的需求必将成为一个热点话题。这将是IT界的革命同时也是挑战，当然对于程序员们来说，这将是一场值得期待的智慧与灵感交融的盛宴。P

（本文取得的成果离不开GMS开发与测试团队的辛勤劳动，特此表示感谢。）



**刘星**  
趋势科技中国研发中心资深QA，在六年职业生涯中，开发和测试的经验恰好参半。2009年加入趋势科技至今，一直从事基于HBase的分布式存储数据库的研发。



**孙靓**  
2005年加入趋势科技中国研发中心。现主要负责趋势科技GMS项目的测试工作。



# 基于分布式Key-Value DB和分布式文件系统的海量云存储系统构建

■ 文 / 高洪

云存储是以数据存储和管理为核心的云计算系统，具备云计算的弹性伸缩、可管理、成本低廉等特征，属于云计算技术的一种延伸。云存储技术的产生有它的必然性，从效率提升、数据容灾备份以及数据中心建设成本等几方面考虑，政府、企业甚至包括个人对云存储系统的集中建设和运营都有潜在需求。云存储技术得到了众多厂商的支持和关注，Amazon推出了EC2和S3，成为了云计算和云存储商用的先行者，同时也建立了云存储即服务（Storage as a Service）的商业模式，目前业界相关云存储标准也在快速完善中。

分布式Key-Value DB和分布式文件系统属于云计算技术的两大基本组件，本文将重点对我们研发的这两种组件的架构、特征和优势、关键技术等方面进行阐述，并介绍基于这两种分布式基础架构的低成本云存储系统的构建方案。

## 分布式Key-Value DB

我们目前研发的分布式Key-Value DB是一款基于Key-Value的高性能、可伸缩、大容量、高可靠性的NoSQL产品，同时具备缓存和持久化存储的能力，对应用提供基于键值对的插入、查询、修改和删除的简单API接口。这款产品不仅可以抵挡互联网应用中海量高并发数据存取的压力，也能够电信级应用中充分发挥优势，为各项目解决数据读写性能瓶颈和海量数据存储问题。

### 架构

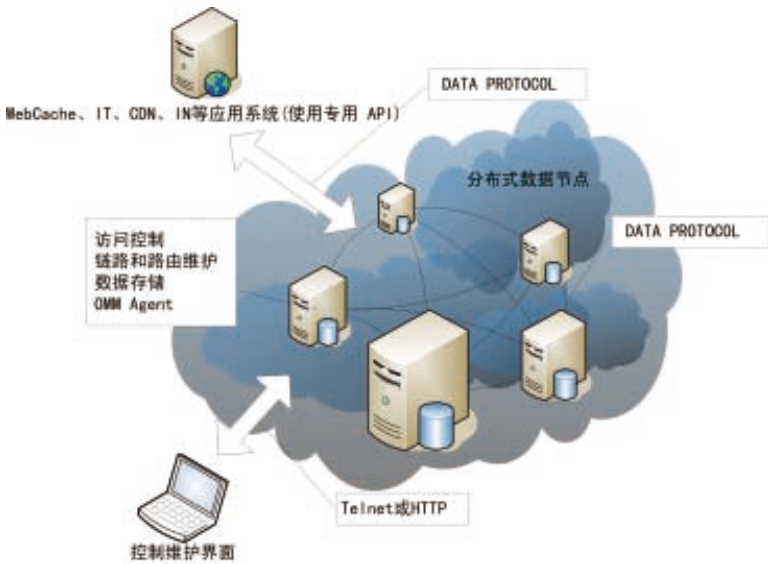
分布式Key-Value DB集群采用无主架构，所有服务器节点地位完全一致，互相之间采用网状的全连接方式。数据在集群各节点均匀分布，集群数据处理能力随集群中节点数量的扩充线性增长。集群通过数据的多副本机制能够提高系统的可用性，某几台服务节点的宕机对应用的数据访问没有任何影响。服务器节点能够根据应用的需求灵活配置数据是否持久化存储。

分布式Key-Value DB同时提供操作控制台，能够登录到任何一个服务节点对集群的成员关系、访问负荷、数据分布进行监控和配置，同时通过操作维护台能够完成分布式Key-Value DB集群软件版本的安装、升级和配置。目前分布式Key-Value DB提供基于命令行（Telnet登录）和基于B/S的图形化运维方式，如图1所示。

### 特性和优势

特性	描述
高性能	存储模式无论采用缓存或持久化引擎，在性能对比测试中皆大幅领先于同类产品
可扩展	可以在线扩展，对应用透明
大容量	因为可以线性扩展，所以集群容量可以很大

图1 分布式Key-Value DB的架构



高可靠	数据多份备份，基于NRW算法的数据最终一致性，保障存储的可靠性
多应用空间	应用空间机制可以使不同应用的数据分别管理，不同的空间分别配置不同的副本、路由、存储策略，数据存储完全隔离
多级存储	支持内存、SSD、硬盘三级存储，有多种组合模式满足应用不同的存储需求
支持大数据的存储	最高支持128MB，客户端进行分包处理
无中心化的配置管理	避免了管理控制台的单点故障
多种控制维护方式	支持基于OMMP的B/S方式管理控制，同时提供Telnet管理控制方式，非常人性化和易于使用

关键技术

■ 多模式的持久化引擎支持

分布式Key-Value DB目前支持以下两种持久化存储引擎。

■ BDB (BerkeleyDB)：以嵌入式K-V数据库方式提供海量数据存储。

■ IFS (Indexed Fast Storage)：基于内存索引的顺序持久化模式存储。

IFS模式的目标是尽量将磁盘的随机写变成顺序写，最大限度地提高磁盘读写的I/O吞吐，IFS模式下文件的创建、修改、删除操作都演化为文件的追加。IFS引擎在内存中存放数据索引，索引永远指向最新的数据，因此一次数据的读取最多涉及一次物理I/O。在系统空闲时，IFS引擎会根据统计信息智能地进行文件整理，清除无效数据，提高磁盘空间的利用效率，清理过程支持断点模式，即清理能够在上次中断的基础上继续进行。

BDB和IFS这两种模式可在同一集群同时使用，但最好使用不同的磁盘，否则性能上会有互相干扰，其中IFS模式尤其适合小文件的持久存储，例如彩信附件、邮件、图片等，在一定程度上可替代NAS。

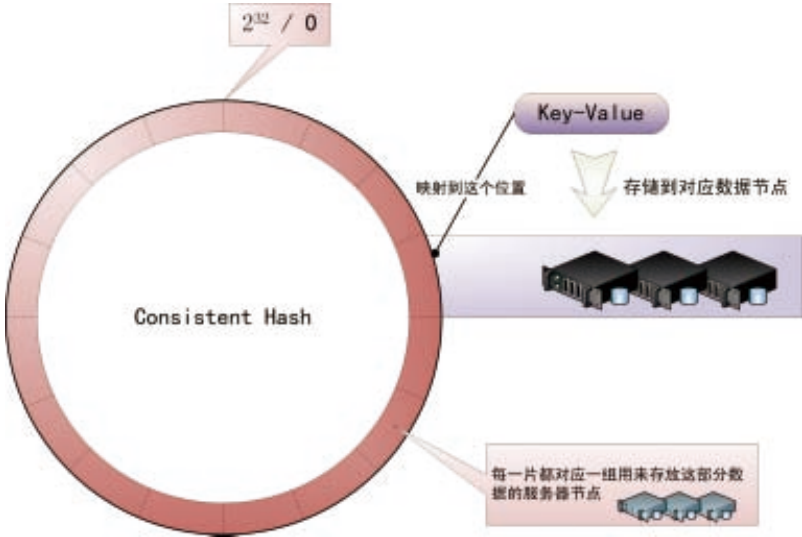
■ 一致性Hash和虚节点

分布式Key-Value DB服务节点采用一致性Hash算法处理数据分布，由于数据节点服务器的机型并不统一，其性能和容量也是不同的，因此可以使一个物理节点负责多个Hash区间的处理，使高端机器能够被充分利用。在出现热区时，可以将过热的Hash区间以虚拟节点的方式放在负荷较低的物理节点上。

分布式Key-Value DB结合了一致性Hash和虚拟节点的特点并加以改进，形成了如图2的方案：将 $2^{32}$ 的Hash空间等分为若干分片，每个分片

即是一个虚节点，根据各物理节点性能差异配置处理不同数量的虚节点，这些虚节点在物理节点上的部署关系即形成虚节点的路由。通过一致性Hash和虚节点相结合的方式，实现了数据在集群的均匀分布以及数据服务器节点热点的消除。

图2 一致性Hash和虚拟节点相结合的改进方案



■ 智能路由交换

路由是指分布式Key-Value DB集群中虚节点在数据服务节点上的分布情况。分布式Key-Value DB构建了一个分布式锁同步系统来存放全局路由表，这张路由表是集群路由管理的基准表，路由变更时必须首先要修改这张路由表中对应的路由记录。为避免每次路由查找都需要查询分布式锁服务，各数据服务节点在本地同时存储全局路由表，路由查找时可直接在本地进行。这样带来的一个问题是本地路由记录可能已经过期，因此在路由记录中通过增加修改时间戳来进行路由记录版本的控制。

路由交换是通过数据访问请求被动触发，同时集群中每个节点的路由管理模块也会定时启动路由交换，通过这种类似病毒传染式的智能路由交换，路由变更能在集群的所有节点中快速生效。客户端API的路由记录也采用同样的方式：客户端API本地缓存路由信息表，在数据访问的同时完成和服务节点的路由交换，大大提高了路由查找的效率，降低了数据访问的时延。

应用场景

我们研发的分布式Key-Value DB集高性能、高吞吐、高可靠性、高扩展性于一身，解决了应用普遍面临的数据可靠性、大容量内存共享、多模块数据一致保障、线性扩容等难题，成为应用云化架构的底层支撑，在传统电信领域和互联网领域都有着广阔的应用空间。

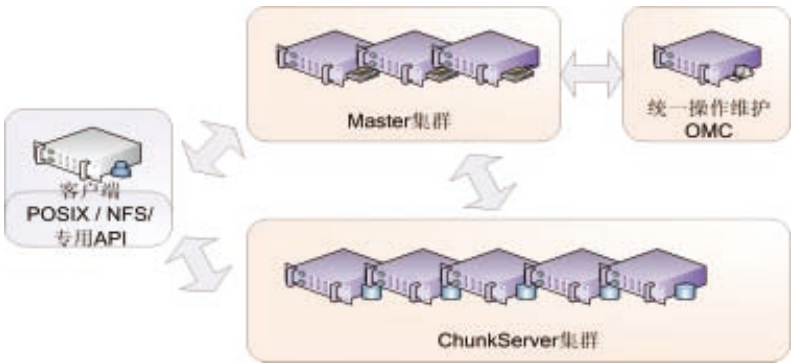
分布式Key-Value DB凭借高效的持久化引擎IFS，最大限度地发挥磁盘I/O的效率，同时结合高效的内存缓存算法进一步提升数据访问的命中率，因此分布式Key-Value DB也能够作为云存储的基础架构，在海量云存储系统的建设中发挥作用。

分布式文件系统

架构

分布式文件系统采用数据和元数据分离的非对称架构，主要由Master、Chunkserver、OMC部件组成，Master提供元数据的高速存储，负责集群内数据存储节点的负载均衡和数据调度迁移。Chunkserver负责文件数据块的存储，对磁盘I/O有较高的要求。OMC是集群的操作维护中心，操作管理员通过OMC即可实现集群运行情况的监控和管理，如图3所示。

图3 分布式文件系统的架构图



特征和优势

特性	描述
高性能	分布式文件系统对大文件进行切片，切片提升文件存储性能，同时分片在硬盘中顺序存放、顺序读写，可以充分发挥硬盘I/O能力
超大容量、良好的扩展性	系统可以通过不断增加Chunkserver实现线性扩展，规模扩展很大后整个系统吞吐量和存储容量都可以做得很大，新加入的Chunkserver主动向Master节点注册

负荷均衡，热点消除	分布式文件系统对大文件进行切片，文件分片均匀分布在多台 Chunkserver上，读写负荷在服务器集群中均衡，另外Master可以通过对多个数据服务器资源占用进行监控，每次选中负载小的服务器进行服务，进一步消除热点
数据可靠	数据多份备份，副本策略可配置。基于Redo日志的元数据备份，数据块校验机制
管理简单	文件系统集群集中管理，减少了设备管理维护人员的投入，系统自动完成内部存储空间的分配、扩容、数据备份、数据恢复，无须对存储内容进行优化，降低了对管理维护人员的技术要求

关键技术点

分布式文件系统主要针对海量大文件存储应用设计，业界主流的设计架构都大致相同，关键技术点集中在分布式元数据管理、分布式目录结构、多接口协议支持、存储节点高效的磁盘管理等方面。

应用场景

分布式文件系统以大文件存储为主，支持少量小文件存储，数据访问采用流式接口，支持标准POSIX和NFS协议。写操作以大的串行写为主，写完后更改少，支持少量随机写。分布式文件系统对数据存储节点的磁盘I/O要求较高，一般会配合专用的存储服务器使用。

对文件切片是分布式文件系统的特征之一，为简化系统设计，通常会采用固定的尺寸对文件进行切片，因此分布式文件系统并不适合同时存储尺寸变化范围较大的文件，分片的尺寸太大会造成小文件存储时磁盘空间的浪费，太小会造成Chunkserver上文件系统句柄（Innode）资源占用太多进而限制了存储的容量上限。

目前分布式文件系统在流媒体存储、监控系统后端存储、高性能CDN集群中有广泛的使用。

优势互补构建云存储

分布式Key-Value DB和分布式文件系统两种组件的主要特性比较如下。

■ 访问接口

分布式Key-Value DB采用Get/Set接口，读写请求一次完成，非常适合小文件的读写。

分布式文件系统采用流式文件接口，一次读写可以分多次调用，更适合流式大文件读写。

■ 数据一致性

分布式Key-Value DB采用无中心化的设计模



式，基于NRW算法实现数据最终一致性。

分布式文件元数据管理由Master集群集中控制，因此能够实现数据的一致性。

■ 可扩展性

分布式Key-Value DB去中心化架构，所有节点地位完全一致，分布式文件系统内部分成Master和Chunkserver两个集群，相比而言Key-Value DB扩展性更强。

■ 性能

分布式Key-Value DB通过智能路由交换，客户端请求能够直接路由到目标数据服务器，分布式文件系统在数据读写时则需要首先和Master通讯获得元数据信息，然后才能进行真正的数据读写。另外分布式Key-Value DB处理负荷由集群所有服务器节点平均分摊，而分布式文件系统的访问性能受限于Master集群的处理性能。因此，同样集群规模，分布式Key-Value DB的读写访问时延更小，并发吞吐更高。

储应用系统构建中都能找到自己的位置。

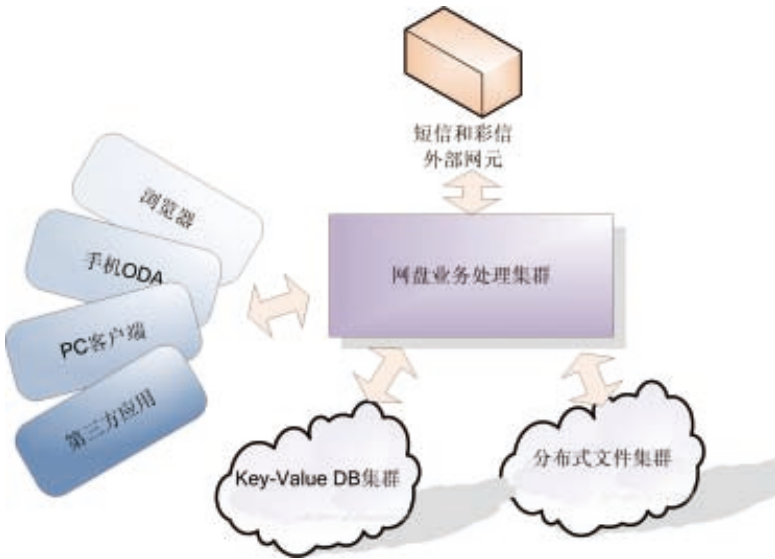
下面结合网盘应用做简单的介绍。网盘属于典型的云存储应用，目前不少厂商已经对公众提供网盘存储服务，图4是网盘系统的架构图，分布式文件系统提供大尺寸文件的存储，比如超过32MB的文件。分布式Key-Value DB集群作为缓存提供用户相关控制信息的存储，提升访问性能，同时还提供小尺寸文件的持久化存储。分布式Key-Value DB和分布式文件系统作为底层存储基础设施，共同配合实现了云存储网盘应用系统的构建。

云存储未来发展趋势

基于软件的低成本云存储已经成为未来存储发展的一种趋势，从技术本身的发展来说，未来云存储将在标准规范、数据安全保障、云存储客户端便携易用等方面进一步完善；同时在政

府、企业和个人对云存储应用需求进一步明确的大背景下，软件云存储技术凭借低成本、易管理等优势，和现有各类数据应用相结合，进一步丰富存储即服务的商用模式，实现快速落地为最终用户带来切实便利，预计我们未来将会看到越来越多样化的云存储应用，以分布式Key-Value DB和分布式文件系统为代表的云存储基础构件将会大有用武之地。P

图4 网盘系统架构图



■ 数据存储

分布式文件系统数据存储基于操作系统的文件系统，存储的块文件数量受操作系统的文件系统句柄数量的限制。分布式Key-Value DB对数据进行了重新的整合，磁盘数据文件数量大大减少。

基于上述不同的特性，分布式Key-Value DB和分布式文件有着各自独特的适用场景，在云存



高洪

中兴通讯业务研究院平台研发总工，中兴通讯技术专家委员会专家。研究方向为IN技术、信令网络、开放业务平台、IMS业务，目前主要从事云计算平台研发工作。

# 淘宝Oceanbase云存储系统实践

■ 文 / 杨传辉

通俗地讲，云计算就是把基础设施以服务的形式打包对外销售，它是一种商业模式，而其中的云存储是技术难点。可以从两个维度分析云存储系统的特性：功能和可扩展性，这是一个“鱼和熊掌”不容易兼得的问题。不同的数据规模，不同的事务和一致性要求，不同的系统故障容忍度，都可能导致不同的存储系统设计。国外的互联网巨头Amazon、Google、Microsoft、Yahoo都有自己的云存储系统，国内的淘宝也研发了自己的云存储系统Oceanbase，并开始应用到联机事务处理OLTP（On-line Transaction Processing）和联机分析处理OLAP（On-line Analytical Processing）业务。

## 云存储系统数据结构

为了保证存储系统的可靠性，需要将数据复制为多份。当数据规模增加时，我们可能会对传统的数据库分库分表以水平扩展，很多企业还开发了各自的数据库中间层以屏蔽分库分表规则。然而，在传统的分库/分表架构下，每一份数据只能为一组Master-Slave节点服务，这就导致同一组机器节点存放了完全相同的数据，当其中某个节点发生故障时，只能通过所在机器组中的节点进行故障恢复，这样的系统称为同构系统。

云存储系统一般指导异构系统，每份数据可以被动态分配到集群中的任意一个节点，当某个节点发生故障时，可以将故障节点原有服务动态迁

移到集群中的任何一台机器。只有实现系统异构才能发挥分布式集群的规模优势，减少集群运维成本，适应云存储系统数据量快速增长的需求。

数据结构决定了云存储系统的功能，云存储系统的数据结构主要有两种：分布式Hash表和分布式B+树，如图1所示。分布式Hash表通过比如一致性Hash的方式将数据分布到集群中的不同节点，数据随机分布，不支持范围查询；而分布式B+树的数据连续存放，支持范围查询，但是需要支持分裂和合并，实现相对较为复杂。

常见的Key-Value系统的数据结构一般为分布式Hash表，只支持基本的Put、Get和Delete操作，比如Amazon的Dynamo和S3系统。而Amazon Simpledb按照domain进行数据划分，规定同一个domain数据量不能超过10GB，从而可以存放到一个数据节点，用户只允许在同一个domain内部执行范围查询操作。Amazon的云存储系统看起来不完美，但相当实用。

Google的系统设计之初就强调可扩展性。从最初的GFS到BigTable，再到后来的Megastore、Percolator，Google先将系统的可扩展性发挥到极致，以后再逐步加入分布式事务、SQL支持等功能。这样的设计得益于Google强大的工程师团队和公司一直以来崇尚通用系统的文化。Google的云存储分为两层：分布式文件系统GFS和分布式数据库系统BigTable，GFS是一个带有追加功能的分布式文件系统，BigTable将事务的提交日志追加到GFS中做持久化。数据在BigTable内连续存储，逻辑上构成一棵分布式B+树，Megastore、Percolator又在BigTable的基础上加入分布式事务、索引、SQL支持等功能。Google的系统设计比较贵族化，可以远观，但模仿前请三思，比如将系统分成多层可能会增加用户操作的延时，对工程师的设计编码能力提出了更高的要求。

Microsoft SQL Azure是一个传统数据库厂商

图1 云存储系统分类图



在云存储系统设计上给出的答案。当数据量增长时，必然要求牺牲部分功能来换取可扩展性，这对于Microsoft是不愿意看到的。Microsoft直接在原有的关系型数据库SQL Server上进行分布式扩展，尽可能多地支持SQL功能，其功能非常丰富，但系统内部不支持SQL Azure实例的分裂和合并。因此，SQL Azure内部也限制了单个SQL Azure实例允许的最大数据量，如Business Edition的最大数据量不超过50GB。相比Google的系统，Microsoft系统的扩展性较弱，但功能较强。

云存储系统的难点在于状态数据的迁移和持久化，状态数据也就是系统的事务提交日志。Google BigTable通过分布式文件系统GFS持久化提交日志，Microsoft SQL Azure直接将提交日志通过网络复制到数据的多个副本，而PNUTS通过Yahoo!内部的分布式消息中间件Yahoo! Message Broker持久化提交日志。Yahoo!没有对外提供云存储服务，但这样的设计可扩展性也是相当不错的。

## 淘宝Oceanbase架构设计

淘宝Oceanbase是从2010年5月开始研发的，其定位是解决淘宝内部在线业务的云存储问题。我们在设计系统时，总是考虑现在及今后一段时间的需求。互联网业务大致可以分为OLTP和OLAP两类，对在线存储的需求简单归纳如下。

- OLTP：今后数据规模为千亿级，数据量百TB，要求几十万QPS和几万TPS。
- OLAP：支持千万级记录的数据集上进行实时计算。
- 功能：支持范围查询，支持跨行跨表事务。
- 其他：5个9的可用性、自动故障处理、自动扩容等。

OLTP和OLAP业务对性能的要求使我们必须采用分布式方案。另外，淘宝的业务发展迅猛，传统的分库/分表方法带来的扩容及运维成本太高，必须构建异构的云存储系统。通过进一步分析在线业务，我们发现互联网在线存储业务有一个特点：数据量虽然很大，但新增数据量比较小，每天新增数据量基本在1TB之内。此外，淘宝的业务面临一些其他挑战，比如需要高效支

持跨行跨表事务，需要支持两张几亿到几十亿条记录的大表进行联表操作。淘宝的海量数据以及复杂的功能需求对存储系统的设计提出了新的挑战，关系型数据库在数据量上有点儿力不从心，而云存储系统又不能高效地支持复杂的功能要求。因此，需要融合关系型数据库的功能和云存储系统的可扩展性这两个优点。

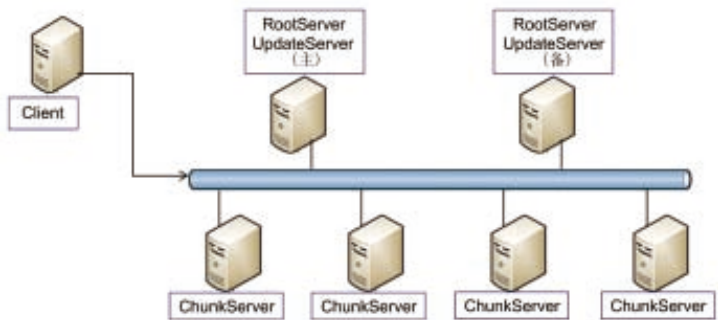
如何借鉴已有技术满足淘宝未来一段时间内的云存储需求？如果直接模仿国外的互联网巨头，比如模仿GFS + BigTable，淘宝的团队确实有一定的经验。然而这样的系统在两年之内很难稳定，并且不能满足跨行跨表事务等复杂的功能需求。既然在线业务新增数据量比较小，那是否可以把最新修改的数据和以前的数据分离呢？

答案是肯定的。淘宝Oceanbase将数据分成动态数据和静态数据两部分：动态数据的数据量较小，侧重TPS和QPS，采用集中式的方法存放到单个节点的高品质存储介质，如内存和SSD；静态数据的数据量很大，侧重存储容量，采用分布式的方法将数据分布到多台普通PC服务器的磁盘或者SSD。由于动态数据的存储介质成本较高，需要不断地将动态数据合并到静态数据中，从而分布到多台机器以实现分布式存储。

淘宝Oceanbase系统架构大致如图2所示。从图2可以看出，系统有以下几个主要模块。

- RootServer：负责数据定位、机器管理、负载均衡、全局表Schema信息管理等。
- UpdateServer：负责存储动态数据，存储介质为内存和SSD。
- ChunkServer：负责存储静态数据，数据存储3份，存储介质为磁盘或者SSD。
- Client：Oceanbase提供的胖客户端。

图2 Oceanbase架构图





写事务只操作UpdateServer，读事务需要同时读取ChunkServer和UpdateServer。某些操作，比如OLAP分析型操作可能需要涉及多个ChunkServer上的数据，这时将引入一个新的MergeServer模块将请求拆分到不同的ChunkServer，合并每个ChunkServer的返回结果后执行排序、分组、分页等操作。静态数据在ChunkServer中保存三份，UpdateServer通过Linux HA的方式进行双机热备以保证可靠性。RootServer的访问压力很小，一般可以和UpdateServer部署在相同节点上，并采用相同的Linux HA方式。Oceanbase的UpdateServer在同一个IDC机房采用实时同步的方式保证强一致性，这意味着写事务只有等到主机和备机都操作成功后才返回客户端。Oceanbase支持跨IDC机房的异步准实时热备，多个机房之间的数据延迟为秒级。

Oceanbase的静态数据和BigTable类似，数据被分为几十到几百MB不等的子表，每个子表的磁盘存储格式为SSTable，通过bloom filter、block cache、key value cache等方式进行优化。SSTable支持根据column group按列存储，从而高效地支持OLAP分析。动态数据采用copy-on-write的方式实现了单机内存中的B+树，在单写多读的应用场景下不需要加锁。

Oceanbase静态数据构成一棵分布式B+树，动态数据为单机B+树。与线下MapReduce批处理应用不同，在线存储应用的更新量一般比较小，动态数据服务器不会成为性能瓶颈。这也就意味着，淘宝Oceanbase用一种更为简便的方式在底层实现了和其他互联网巨头类似的B+树数据结构，并且能够高效地支持跨行跨表事务。当然，当数据量增长到万亿级或者数据更新更快时，需要考虑将动态数据服务器的方案由集中式修改为分布式。我们也考虑过多UpdateServer方案的设计，但由于短期内看不到明确的需求，暂时没有实现，目前我们认为可以通过硬件的方法，比如万兆网卡、更好的CPU、更大的内存和SSD来解决。

Oceanbase还实现了一些分布式系统中常见的特性，比如自动负载均衡、在线修改Schema、内置压缩解压等。另外，Oceanbase系统里面没有随机写操作，因此天然适应SSD存储介质，很好地弥补了磁盘的IOPS不足这个问题。

## Oceanbase应用效果和经验

Oceanbase首先应用在淘宝收藏夹并取得了明显的效果。淘宝收藏夹最初采用MySQL分库/分表的方式实现，通过使用Oceanbase，机器数由原来的16台主加上16台备共32台减少到12台静态数据服务器加上2台动态数据服务器，大大节省了机器资源。另外，目前应用的很多问题在Oceanbase中是通过更好的架构来解决，单机层面基本没做优化，相信后续还有很大的提升空间。

在这过程中，我们也积累了一些经验教训。

■ **选择合适的技术。**云存储听起来比较神秘，但实际上，对于大多数企业，需要设计好系统可扩展性发展的路线图，当数据规模比较小，可以采用传统的分库分表的方式构建同构系统；当数据规模逐步增加时，可以考虑构建符合企业需求的异构系统。

■ **细节决定成败。**云存储更多地是一个工程问题，代码质量、优化细节对系统的表现影响至关重要，淘宝Oceanbase的大多数代码都被两个以上的工程师Review，我们也在减少Cache锁粒度、减少上下文切换、减少内存分配和内存拷贝等方面做了很多细粒度的工作。

## 展望

Oceanbase目前的主要工作是应用推广，根据应用的需求来逐步完善Oceanbase系统，实现互联网数据库的构想。我们已经开始和淘宝的业务团队开展了千万级数据秒级实时分析的OLAP项目。另外，Oceanbase还在考虑整合分布式Blob存储系统。随着应用推广的深入和Oceanbase系统的优化，希望能在合适的时间进行数据库新基准TPC-E的测试。

另外一个振奋人心的消息是：Oceanbase将在合适的时间点开源。相信通过业界同仁一起努力，一定能够将云存储这个问题解决好！



杨传辉

花名日照，淘宝存储系统专家，热衷于分布式存储和计算系统设计，对分布式系统理论和工程实践有比较深厚的理解。之前在百度作为核心成员主导或参与MapReduce、BigTable和分布式消息队列等底层基础设施架构工作。

# 寻找机遇 创造未来 —— CSDN 热门职位全新推荐

## 北京搜狐新媒体信息技术有限公司

这里有广阔的平台——我们创造无数辉煌，并保持引领，不断超越；

这里有无尽的创意——我们坚信技术驱动世界，创新改变未来；

这里有优秀的文化——我们倡导诚信、公正、包容、证明、沟通；

这里有年轻的力量——我们拥有热情和梦想，生命在此绽放；

搜狐公司海量技术职位火热招聘中，诚邀精英加盟，了解职位请登录 <http://events.csdn.net/sohu.pdf>，更多职位请查询 <http://hr.sohu.com/>。



## 搜狐畅游



### 游戏研发部3D程序专员

工作职责：

1. 3D 框架下各种功能模块和算法的实现；
2. 研究、开发和优化3D引擎。

职位要求：

1. 诚信正直，具备高度的责任心；
2. 计算机相关专业本科及以上

学历；3. 两年以上的3D引擎开发经验；4. 精通C++，面向对象编程，Windows编程；5. 熟悉3D图形学原理，熟悉3D几何；6. 精通DirectX/OpenGL，精通DX9；7. 具备良好的算法和数学基础，优秀的编程习惯和学习能力。

公司网址：<http://www.changyou.com> E-mail: [gamejob@job.cyou-inc.com](mailto:gamejob@job.cyou-inc.com)

## 上海科锐福克斯人才顾问有限公司

**Creerfocus**  
高薪猎聘IT人才

**Careerfocus**  
科锐福克斯

上海科锐福克斯是一家综合人力资源服务专业顾问公司，我们像所有企业雇主一样认同：人才类似于资本，是企业的核心资源之一。我们的服务是通过专业的工作为企业客户发现、寻访并协助甄选出合适的人才，来完善企业的人力资源结构，支持客户的经营战略能够有效实现！

联系方式：0755-26014787 网址：[www.careerfocus.com.cn](http://www.careerfocus.com.cn)

## 欧特克软件（中国）有限公司

**Autodesk** 你一展身手的舞台，  
打造美好未来

你一展身手的舞台，打造美好未来加入我们的队伍，以一种全新的方式实现设计理念。无论是设计可持续性建筑还是开发混合动力汽车，Autodesk软件都可助你一臂之力。作为全球领先的二维与三维设计、工程和娱乐软件提供商，我们致力于帮助打造出更美好的未来。

让我们共同创造未来！

更多职位信息请浏览网站：[www.autodesk.com/careers](http://www.autodesk.com/careers)  
或直接将简历发送至：[acrd.hiring@autodesk.com](mailto:acrd.hiring@autodesk.com)

## 方正国际软件有限公司

**急聘以下岗位**

JAVA软件开发工程师  
MIS软件开发工程师  
Delphi软件开发工程师  
Unix/C软件开发工程师

**FOUNDER 方正**

C++软件开发工程师  
HIS/CIS系统设计开发人员  
Unix高级系统工程师  
开发总监

详细信息请登录智联，前程无忧或中华英才查询，并投递相应邮箱！

公司主页：[www.founderinternational.com](http://www.founderinternational.com)

公司地址：北京市海淀区北四环西路52号中芯大厦5层

电话：0512-86665500-6506 邮箱：[srhr@founder.com](mailto:srhr@founder.com)

## 杭州引力网络技术有限公司



职位描述：

负责从事 iPhone/iPad/Mac 平台应用程序的开发及测试；

**iOS Developer**

任职要求：

1. 有半年以上iOS/Objective-C、Cocoa、XCode使用及开发经验；
2. 良好的编程习惯，熟悉iOS App开发流程；
3. 参与或独立进行过一次完整的iPhone/iPad App产品开发；
4. 具有团队合作精神 and 沟通技巧；
5. 喜欢并有志于长期从事编程工作，喜欢研究和新技术；
6. 有AppStore线上作品者优先，有Linux开发经验者优先。

公司网址：[www.codans.com](http://www.codans.com) 邮箱：[hr@codans.com](mailto:hr@codans.com)

电话：0571-87770666 18005710666 联系人：胡先生



## 亿阳信通股份有限公司



**诚招以下职位**

软件项目经理、资深技术经理、设计组长、中高级Java开发工程师、C++/C#开发工程师、需求分析师、DA分析师、流程开发工程师、实施工程师、测试工程师、UI设计师等职位。

**欢迎您的加入！**

公司网站：<http://www.boco.com.cn:8080/bocoit/>

公司地址：北京市海淀区杏石口路99号西山赢府AB座

简历投递邮箱：[zhaopin@boco.com.cn](mailto:zhaopin@boco.com.cn)；[lily@boco.com.cn](mailto:lily@boco.com.cn)

## 上海纵游网络技术有限公司

我们正在全力招聘中国人才，打造一个全新的中国团队，致力于中国手机游戏平台的构筑和运营。愿意加入我们，一起构筑世界顶级的手机游戏平台吗？

即日起，通过本广告于2011年6月30日前入读者，在试用期结束时即送iPad2一台！Android/iPhone/Web application，更多职位招聘信息请登陆公司主页进行查询。

公司主页：[www.denachina.com](http://www.denachina.com)

**DeNA China**



请以“职位+姓名+CSDN”作为应聘邮件标题，  
将中英文或中日文简历发至[hr@denachina.com](mailto:hr@denachina.com)

# 有向无环图——Dryad

■ 文 / 徐永睿

Dryad是Microsoft的分布式并行计算基础平台，不同于Google的MapReduce，Dryad主要用于Windows HPC Server平台上的大规模并行计算。本文将结合Microsoft的技术论文*Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks*重点讲述Dryad采用的有向无环图并行编程模式。

## 有向无环图并行编程模式的优势

Dryad与MapReduce的设计目标基本一致，主要是解决大规模静态数据的批量处理，比如分析系统日志等。但这两者背后的编程模型却区别较大，MapReduce的灵感主要来源于LISP等函数式编程语言中的Map和Reduce操作。在MapReduce中，开发者只需要根据应用需求关注系统特定的Map和Reduce操作，几乎不需要关注分布式处理的诸多细节问题。同时，由于MapReduce系统执行过程、系统通信方式基本固定，MapReduce的控制能力稍显不足，灵活性也有所欠缺。因此，MapReduce简化的编程模型使其更适合于那些没有多少并行计算经验的开发者方便快速地进行并行应用程序开发。

Dryad最初的开发动机就是允许开发者对系统的交互过程、并行化等进行精细的控制。因此，Dryad采用了灵活的有向无环图并行模型。在这个有向无环图中，每一个顶点代表系统需要完成的一个任务，而顶点之间的边表示两个任务之间的通信。由于每一个顶点代表的任务都是一个完成单一功能的简单任务，内部并不需要进行并行化，业务逻辑实现非常容易。同时，Dryad允许开发者对有向无环图中的每一条边进行精细控制，任务之间的数据传输机制既可以采用共享内存，也可以采用文件、TCP管道通信等，相对于MapReduce中Mapper和Reducer之间单一高时

延的RPC通信方式，Dryad灵活的通信机制明显适用范围更广而且能够取得更为理想的性能。此外，Dryad可以通过组合一些简单的有向无环图构成更为复杂的图，在不明显提升应用开发复杂性的前提下就能简单地适应具有复杂并发关系的应用程序。MapReduce对常见的数据集之间的联接操作支持非常有限，无论是Map端还是Reduce端的联接操作都需要开发者技巧性的实现，否则只能通过采用构建于MapReduce之上的Pig、Hive或Sawzall等提供支持。而在Dryad中，由于有向无环图支持任意数量的输入和输出顶点，支持SQL风格的联接操作非常简单，同时，Dryad单一Job支持多输入多输出数据集也和MapReduce中一个Job只能对应单一输入数据集和单一输出数据集形成鲜明对比。在MapReduce中，当涉及不同数据集时，往往需要采用多个Job以管道的形式完成最终任务，这种方式虽然很好地符合了“Do one thing and do it well”的编程哲学，但对于一些简单的任务而言，稍显“笨拙”。

## 从SQL查询说起

SQL查询是我们每天不得不大量面对的操作，其中需要联接的查询占了绝大多数。MapReduce恰好对这种常见数据集联接操作支持不佳，而Dryad通过有向无环图给出了相应的解决方法。下面结合论文中提到的SQL查询例子介绍Dryad采用的有向无环图并行编程模型。

论文的SQL例子中，有两张表，第一张表为photoObjAll，每一条记录存储一个天体对象，其主码是ObjID。另外记录中存储了该天体对象颜色的各亮度分量的对数大小，这些亮度分量在表中的属性分别是u、g、r、i、z，mode属性为1表明该天体对象是“Primary Object”。第二张表为neighbors，存储了1个天体对象在30弧秒（角



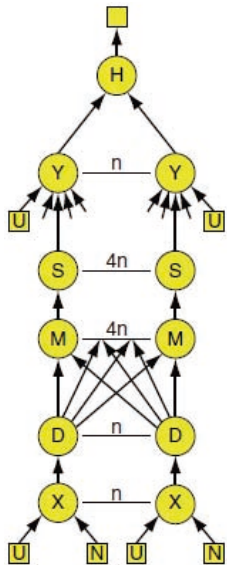
度单位)内的相邻天体对象,因此有objID和neighborObjID这两个属性。该查询的目标是找到所有天体对象,满足该天体对象存在至少一个颜色相近的相邻天体对象。SQL代码如图1所示。

图1 找到所有拥有相近颜色相邻对象的天体对象的SQL代码

```
select distinct p.objID
from photoObjAll p
join neighbors n — call this join “X”
on p.objID = n.objID
and n.objID < n.neighborObjID
and p.mode = 1
join photoObjAll l — call this join “Y”
on l.objid = n.neighborObjID
and l.mode = 1
and abs((p.u-p.g)-(l.u-l.g))<0.05
and abs((p.g-p.r)-(l.g-l.r))<0.05
and abs((p.r-p.i)-(l.r-l.i))<0.05
and abs((p.i-p.z)-(l.i-l.z))<0.05
```

上述代码非常直观,查询过程经历了两次表联接操作,分别称为X和Y。为了加速查询过程,对两张表中该查询涉及各列分别建立索引。在案例中,只提取两张表中每条记录里索引所对应的列的数据,到两个与索引相同顺序排序(索引均以objID为键)的二进制文件“ugriz.bin”和“neighbors.bin”中。这样,消除了两张表中与该查询无关的属性,而这两个二进制文件将会

图2 该SQL查询对应的有向无环图



作为Dryad中的输入。图2就是该SQL查询在Dryad中所对应的有向无环图。

$U_i, N_i$  ( $i$ 从1到 $n$ )表示的是两个二进制文件按照objID区间范围等分成 $n$ 份的输入片段。同时,在X联接中,由于是以 $p.objID = n.objID$ 作为联接方式,而两个二进制文件在产生时均以objID排序,那么 $U_1$ 中所包含的objID的范围一定和neighbors表 $N_1$ 片段所对应,即photoObjAll表 $U_1$ 片段中存在的objID的联接目标不会出现在除 $N_1$ 之外其他 $N_i$ 片段中。这样,将文件分片后各个片段之间并无关联,如同MapReduce的Map阶段,实现了扩展性极高的数据并行模式。当需要处理的数据量增大时,只需要简单地增加更

多的X顶点即可。

X联接处理完后,将X联接完毕的数据通过X与D之间的边传递到顶点D,D的任务是X联接后的记录信息重新按照neighborObjID区间范围进行分发。之所以重新按不同属性分发,是因为在Y联接中,以 $l.objID = n.neighborObjID$ 进行联接,同上段所述,需要保证联接时各个片段间互相独立。由于作者使用 $n$ 台服务器,每台服务器采用4核处理器,为了在Merge顶点之后的Sort顶点尽可能利用所有计算资源以neighborObjID进行排序,这里将数据分为 $4n$ 个片段。当发生Y联接时,因为 $U_i$ 是按照objID区间范围 $n$ 等分的片段,则S阶段的 $4n$ 片段必须重新合并成 $n$ 份,例如 $S_1$ 到 $S_4$ 与 $U_1$ 进行联接。Y联接之后,将符合约束条件的objID传递给H顶点,H顶点执行对应于SQL中的distinct操作,得到最终的输出。

在本例中,通过将SQL转换成对应的有向无环图,实现了在 $n$ 台4核服务器上的并行化查询处理。从图2可以看出,在单一Job中Dryad可以随时引入新的输入顶点,而如果采用MapReduce,则只能通过多个使用Job的方式进行处理。

## 有向无环图简介

在Dryad中,最基本的组成元素就是有向无环图,其定义为 $G = \langle V_G, E_G, I_G, O_G \rangle$ ,分别代表顶点、边、输入和输出顶点,其中 $I_G \subseteq V_G, O_G \subseteq V_G$ 。在Dryad中,所有的有向无环图由一系列子图构成,而这些简单的子图则是由图3中的这些简单操作所组成。构成有向无环图中所有顶点的程序都继承自Dryad库中的一个C++基类,每一个顶点通过调用合适的静态工厂创建。

有向无环图中最基本的操作就是顶点的克隆操作( $\wedge$ ),如图3(a)、(b)所示,分别将A、B顶点克隆 $n$ 次。这个操作通过定义多个顶点完成相同任务实现了Dryad程序中的并行化,在图2的SQL示例中,除H外的每一个任务顶点都使用了该操作克隆了至少 $n$ 次实现该阶段任务的并行化。不难理解,Dryad的有向无环图整体上实现了一种任务并行模式,而每一个任务在局部上则采用数据并行模式。Dryad为了简化Job的管理,也是采用“Job Stages”来描述单一Job的多

图3 Dryad有向无环图基本操作

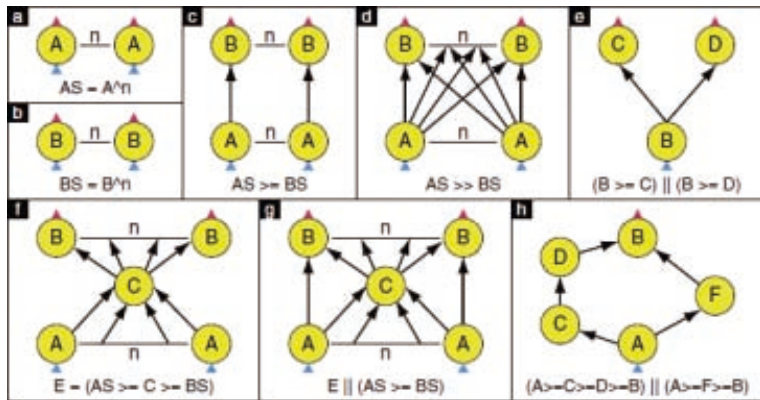
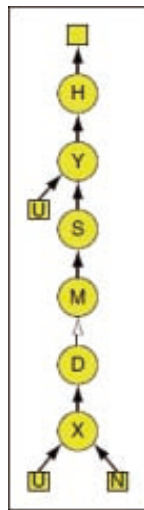


图4 SQL案例对应的Job Stages



个任务而不考虑单一任务顶点多次克隆时的数据并行，图2对应的“Job Stages”，如图4所示。

单一任务顶点可以采用克隆实现并行化，但当不同任务顶点需要进行通信时，则需要图3 (c)、(d)的“>=”、“>>”操作符。

“>>”如图3 (d)所示，用于构建完全二部图（即A中的每个顶点都与B的每个顶点有且仅有一条边相连）。而“>=”操作则会根据A、B任务克隆的顶点数决定实际生成的有向无环图。若A、B克隆顶点数相等，即 $|O_A| = |I_B|$ ，则是图3 (c)所示的一一对应；而如果 $|O_A| > |I_B|$ ，则将所有A顶点的输出以循环方式作为所有B顶点的输入，比如图3 (f)中任务A、C之间，反之亦然，比如图3 (f)中任务C、B之间。

另外一个比较重要的基本操作就是“||”，通过将简单的图合并成复杂的图，用于提升Dryad的灵活性，比如图3 (e)、(g)、(h)。以图3 (h)为例，使用合并操作就轻松地实现了Fork/Join并行模式。

通过上述这些基本操作，就能够轻松地实现构造图2所对应的有向无环图的代码，如图5所示。

## 有向无环图之边（Channel）

在Dryad中，顶点之间的边构成了多个任务间的通信机制，由于可以由开发者进行精细控制，因此具有非常高的灵活性。在MapReduce中，当进行Shuffle过程时，Mapper将任务输出写到自身环形缓冲区，输出数据过大时进而溢写到磁盘文件，Reducer再通过少量复制线程以RPC

图5 图2对应的Dryad构造代码

```
GraphBuilder ISet = moduleX"N";
GraphBuilder DSet = moduleD"N";
GraphBuilder MSet = moduleM"(3*4);
GraphBuilder SSet = moduleS"(3*4);
GraphBuilder YSet = moduleY"N";
GraphBuilder HSet = moduleH"N";

GraphBuilder IInputs = (ugriz1 >= ISet) || (neighbor >= ISet);
GraphBuilder YInputs = ugriz2 >= YSet;

GraphBuilder XToY = ISet >= DSet >> MSet >= SSet;
for (i = 0; i < 3*4; ++i)
{
    XToY - XToY || (SSet.GetVertex(i) >= YSet.GetVertex(i/4));
}

GraphBuilder YToH = YSet >= HSet;
GraphBuilder HOutputs = HSet >= output;

GraphBuilder final = IInputs || YInputs || XToY || YToH || HOutputs;
```

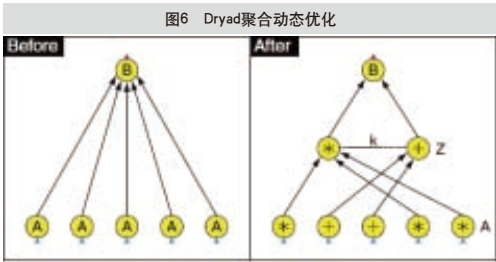
方式得到Mapper输出。虽然整个Shuffle过程都无需开发者干预，但是这种通信机制的开销相对较大，也不够灵活。虽然Dryad在默认情况下也是采用这种生产者输出到本地文件、消费者远程获取文件的方式，但与此同时，Dryad还为开发者提供了TCP管道、共享内存FIFO等通信方式。当顶点与顶点间采用共享内存通信时，虽在逻辑上它们仍然是独立的顶点任务，但在物理实现上它们共存于同一物理进程之中。这里仍以图2为例，在记录信息重新按照neighborObjID区间范围进行分发时，由于D、M、S这三者是完成该任务的流水线上的3个步骤，可以将Di、M4i-3到M4i、S4i-3到S4i（i从1到n）这一组顶点间的通信采用共享内存方式置于同一物理进程中，从而提升系统运行效率。总之，Dryad很好地利用有向无环图边的机制，将任务间通信策略分离。

同时，Dryad的灵活性也是一把双刃剑。虽然“有向”、“无环”这两个特性最大程度地避免了死锁，但如果两个顶点间的边采用的是管道通信方式，则两个顶点必须都已经被并发调度执行，否则，如果Job中顶点数目远大于集群中可调度计算资源，则有可能出现其中一些顶点被调度而与之管道通信对应的顶点未被调度，造成系统死锁。在MapReduce或者完全采用Dryad默认的通信方式，由于任务输出到文件，无需生产者、消费者同时运行，因此不会存在该问题。

## 有向无环图之运行时优化

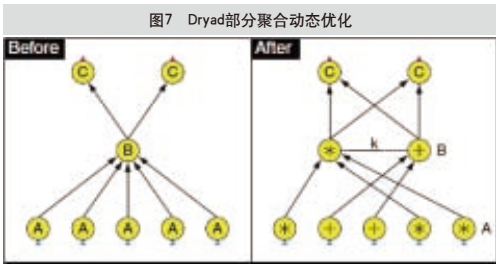
在Dryad中，如果一个Job拥有足够多的顶点而每一个顶点完成的任务又相对简单，那么有向无环图中边与边之间的大量通信就会占据Job执行时间很大比例。分布式系统的一个基本原则就

是计算尽可能靠近数据。MapReduce中JobTracker会将Mapper尽可能调度到离数据最近的节点，而Dryad中每一个顶点也会尽可能被调度到数据所在的服务器或机架上。同时在Job执行过程中，Dryad采用了一种聚合树的优化手段，尽可能减少边与边的通信代价。该思路如图6所示。



该聚合优化思路其实比较简单，在实际Dryad集群中，有些顶点与顶点之间的通信可能代价非常昂贵，比如不在同一服务器、不在同一机架甚至可能不在同一数据中心内。在该优化思路中，通过在原来的有向无环图中引入新的层次，对网络中逻辑距离相近的顶点分为一组，对这些同组顶点的输出进行聚合后再进行进一步操作，从而降低网络访问延迟。

在图6的基础上，Dryad还提供了一种运行时的部分聚合优化，如图7所示。

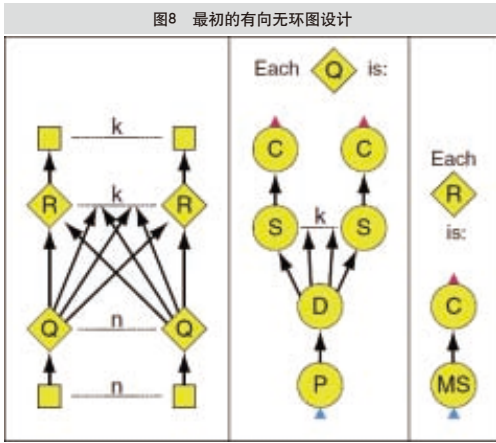


对比图6，部分聚合并非在A与B顶点间增加新的一层顶点，而是对B顶点采用克隆操作进行复制。相对于改进前，每个B顶点只与同组A顶点进行相对较低代价的通信，同时，单一B顶点只需要处理同组A顶点输出数据，计算量也相对减少。数据最终聚合任务交予后续顶点进行处理。在实际中，该优化非常有效。

## 运行时优化案例

这个例子是一个典型的数据挖掘任务，任务的目标是从MSN搜索服务产生的日志中，提

取出用户搜索的关键字，根据关键字出现频率构建直方图。针对该任务，可以构建如图8的有向无环图。



最初的设计如图8左边部分所示，而图8中间和右边部分是Q和R的子图，在设计中，Q和R子图内的顶点均采用共享内存通信。因此，图左边的每一个Q和R对应一个物理进程。如果大家对MapReduce比较熟悉，其实应该能很快反应出该有向无环图是近似等价于MapReduce的执行流程的。Q可以简单视为Mapper而R可以视为Reducer，即n个Mapper任务，k个Reducer任务。在MapReduce中，每一个Mapper和Reducer也是工作在独立的进程之中。图8 Q顶点的子图中，P顶点读取分配给该Q顶点的日志部分，D顶点根据读取到的日志中的每一个查询关键字的Hash分发成k份，然后通过S顶点在每份中对关键字进行内部排序。C顶点完成每个关键字出现次数的累加，C等价于MapReduce中的Combiner，把关键字变为{keyword, numbers}而不是若干个{keyword, 1}的形式，从而降低网络传输量。而R顶点合并输入，再次完成关键字累加，然后输出。

从图8的有向无环图可以看出，这种设计是非常简单、清晰的，但在面临大数据集上，该方案在系统可扩展性上存在一些问题。

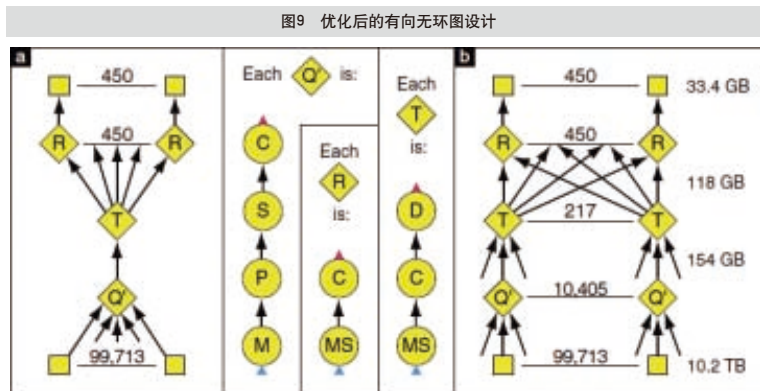
一方面，对于日志的一段分片对应一个Q顶点进行处理的思路，这一段分片大小设置成多大是值得考虑的一个问题。以Hadoop为例，每一个Map操作只处理一个输入分片，而通常情况下，输入都来源于HDFS。HDFS默认块大小是64MB，如果每一个Mapper处理的输入分片大



小等于HDFS块大小, 不管Reducer数目是多少, Mapper中排序的总数据是64MB, 远低于集群中单一服务器的可用内存, 资源无法充分利用。对此一种解决办法是通过在单一服务器上增加同时运行的Mapper数目, 但与此同时, 单一服务器CPU并发计算能力有限, 大量并发Mapper无法真正完全并行执行。而如果增加Mapper处理的输入分片大小, HDFS块大小在不变的情况下, 则情况更为糟糕, Mapper进行Map操作时会操作很多不在本地的输入文件块, 导致性能恶化。

另一方面, 在面临大数据集处理时, 往往图8中的n值会非常大, Q与R之间的完全二部图将会导致系统执行时, 每一个R面临大量的并发输入, 而JobTracker在执行Reduce任务时并没有考虑到数据的本地化。同时, 为了限制Job最终输出文件数目, k往往又远小于n。这样, R会形成瓶颈, 延长系统响应。

针对上述问题, 可以利用有向无环图直观的特点, 结合图6、图7Dryad运行时优化特性, 提出图9的改进型有向无环图设计。



在图9中, 利用图7中的Dryad提供的部分聚合优化特性, 并不采用一个Q'顶点对应一个输入分片, 而是将本地计算机所有输入分片进行部分聚合。同时, 对单一Q'顶点处理的数据上限进行设置, 比如1GB。通过这种方式, 能够充分利用本地可用内存资源又不至于单一顶点负荷过重, 而且所有输入分片都为本地化操作。因此, Q'内部的有向无环图必须先对若干本地输入分片进行不确定合并(M)。为了避免上面所提到的R面临大量的并发输入问题, 在图9优化后的设计中, 引入了新的有向无环图顶点T层次。顶点T利用了Dryad的聚合动态优化, 运行

于同一个交换机或机架中的Q'顶点的输出合并到同一个T顶点中, 降低了网络数据传输延迟。然后在T顶点, 才利用关键字Hash值, 将不同查询关键字分发到不同的R顶点中进行处理。由于顶点T数目远远小于图8中的顶点Q, 因此R顶点所面临的通信Channels大大减少。

除此而外, 改进后的有向无环图通过引入额外的抽象层进行分层设计, 减少了每层顶点程序的复杂性。例如, 相对于图8中的Q, 图9中的Q'将分发职责分离到T中, 子图Q'复杂性大大降低。

通过该优化实例, 利用Dryad提供的运行时优化机制, 结合有向无环图自身设计上面的灵活性, 可以实现高效率的应用程序。

## 总结

关于Dryad, 还有许多分布式处理的诸多问题本文没有论述, 比如Dryad系统结构、作业执行流程、故障容忍策略等, 而构建于Dryad之上的DryadLINQ本文也没有提及。大家感兴趣的话, 推荐阅读Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks和DryadLINQ: A System for General-Purpose Distributed Data-Parallel Computing Using a High-Level Language两篇论文。虽然Dryad有向无环图模式灵活强大, 但目前Dryad尚不成熟。例如, 只针对运行Windows HPC Server的集群设计, 只支持单一Job同时运行, 缺乏Hadoop中日趋完善的作业调度器, 同时也存在类似于MapReduce的JobTracker单点故障(Yahoo!计划重构Hadoop, 能一定程度缓解该问题)。相信Dryad在今后不断成熟过程中会逐步减少上述问题, 从而利用强大的有向无环图并行编程模式, 提供给开发者更多选择。P



徐永睿

系统架构设计师、系统分析师、软件设计师。现在西南科技大学从事并行计算、分布式存储方面的研究。关注函数式编程语言, 经常参与Topcoder算法比赛(Single Round Match)。

# 云计算创业者说

云计算，这种正以指数级速度增长的技术，已经渐渐渗透到IT界的每一个角落，深刻影响着人们的生活。一些敏锐的技术人员不甘寂寞，准确地挖出了潜伏的创业机会，踏上了创业的征程。本刊记者近日采访了三位云计算创业者，请他们分享自己的创业秘籍。



郭欣

监控宝创始人，《构建高性能Web站点》作者。新浪微博：guoxin。

## 创业是与生俱来的情结

**记者：您是怎样走向创业这条路的？**

郭欣：一直以来，我都感觉创业是一种与生俱来的情结，有的人骨子里就将创业视为生活，我可能就算这种人。并不需要什么都懂才开始创业，但一定要有一种敢想敢做的精神。创业是一种特质，能渗透到你的血液中；创业也是一种宿命，你得放弃一些东西。我不记得第一次萌生创业念头的时，因为萌生念头和实际创业是两回事。想想那大概是初中第一次见到WPS的时候，当时或多或少就知道自己未来也得折腾类似的东西了。高中热衷游戏之后，也想自己开发游戏；看到国内没有Web架构和性能优化类的原创书籍，就自己写一本。不明白为什么总是有想自己搞一个的念头，大概因为自己的梦只能由自己来实现。

**记者：您如何看待云计算所酝酿的商机？**

郭欣：看待任何商机，首先要看能否给你的用户带来价值，以及所处行业的发展趋势，而云计算只是一种实现商业价值的工具。因为我们针对的是企业用户，所以用户价值更多体现在能否帮助用户降低成本、提高效率。如果脱离了具体行业 and 用户价值，谈论云计算毫无意义。结合行业来看，我认为国内有很多机会来做SaaS，比如监控宝是针对网站性能监控领域的云服务产品。我们的监控宝是首款SaaS产品，目前取得了不错的成绩，正在保持稳定的业绩增长。虽然与其他面向最终用户的应用相比，监控宝的市场容量相对有限，但对于一款轻量级的SaaS应用来说，这是一个足以立足的切入点，有可能带来更多探索SaaS的机会。

**记者：您认为自己在创业方面的最大优势是什么？发展过程中，有哪些印象最深的事情？**

郭欣：因为我个人做过技术和产品，也

懂性能优化，同时我们团队也在技术方面有一定的积累，所以我们的最大优势在于懂用户，并且用心做产品。我觉得这是创业最重要的前提。懂产品归根结底是要懂用户，因为产品只是一种工具，服务才是目的，服务的目的还是要给用户带来真正的价值，如果不懂用户，一切都毫无意义。监控宝的用户就是技术、运维工作者，因此我自己在相当长时间内都充当产品设计师的角色，首先要设计出自己能够满意的产品。

监控宝创立之初，我们就坚持一个原则，那就是让产品简单，而且是无与伦比的简单。所以我们在产品规划、交互设计以及定价模型上都倾注了大量的心血。我们有很多用户过去都在使用其他监控系统，但他们最终选择了监控宝，理由就是我们的产品足够简单实用。事实上，简单可以带来数不清的好处，但同时也会考验你的坚持，会有一些用户抱怨没有他所需要的功能。有声音是好的，但你必须有足够的判断力和直觉，你要坚持原则，因为你提供的是SaaS，而不是定制软件；你要对所有的用户负责，而不能为了取悦个别用户而给大部分用户制造噪音，因为SaaS依靠的是长尾效应，而不是28法则。

**记者：从技术创业到管理公司，您个人的思想是怎样转变的？**

郭欣：技术和管理都不是创业者最头疼的，归根结底，我们最头疼还是经营问题。这种转变需要一个过程，也是个人商业感觉的不断提升，公司的成长便是最好的驱动力，所需要弥补的知识也是不容忽视的。创业者的学习能力必须非常强，借助互联网你可以让自己不断成长，同时也要借助团队和各种资源的力量，这些都是转型过程中必不可少的。P



范渊  
安恒信息技术有限公司总裁

## 面对困难需要坚持和专注

**记者：您第一次创业的念头是什么时候开始的？是什么原因促成了这个念头？**

范渊：2000年开始我在硅谷的ArcSight公司做信息安全，从最开始的核心研发工作做到后来的主管。因为硕博课题正好涉及Web安全，所以我意识到Web安全、数据安全市场的潜力非常巨大，并且有很好的趋势，所以在2006年我就产生了回国创业的念头。同年，我和几个朋友选择在杭州的滨江高新区开始创业。滨江高新区的建设和布局跟硅谷还是有点儿像的，不是很拥挤，但是缺点是杭州的信息安全人才相对于北京、上海、广州这些城市来说不占优势。我还算比较幸运的，在公司发展的这五年中，不管是巧合，还是因为我们的努力，很多人能够聚在一起。我觉得这点很重要。现在公司业务也在慢慢地做大，发展也很快。另外我想说的是每个公司初创的时候都会遇到很多困难，当遇到困难时首先要坚持，还需要足够的专注，机会还是会向于努力者倾斜。

**记者：您认为云计算的到来对于信息安全意味着什么？**

范渊：云计算的发展应该说是给信息安全带来了一个巨大的机会，哪怕先抛开公有云的概念，我们自己的专网，我们自己的网络，我们自己的在线系统，可能都存在很大的问题。那么现在呢？我们会以一个云的方式来进行相关的资源调用、相互通信、数据分散等问题，云本身的安全问题就显得尤为重要，因为数据有可能会存储在不同的云中，所以保证云中数据的隐私性以及不被篡改才能保证数据安全。在我看来这些因素对我们是一个很好的发展契机，因为我们所专注的Web安全和云安全本身就是紧密联系的。

**记者：相对于其他行业，信息安全并不是最赚钱的，您认为安全行业的创业需要注意什么？**

范渊：安全行业肯定不是最容易赚钱的。站在整个中国投资者的角度来看，很多投资者也不会选择这个行业进行投资。安全行业的创业需要注意两个方面：要足够专注、将社会责任感和经济价值充分结合。我们非常看中社会责任感这方面，像北京奥运会、上海世博会的信息网络安全保障工作，我们都参与并贡献了自己的力量，我觉得参加这些活动本身对安恒就是一个很好的提升。

**记者：从做技术到管理公司，您都遇到哪些挑战？您的创业初衷有变化吗？**

范渊：对我来说，管理确实是一个不小的挑战。就像你说的，我本身走的是技术路线，现在的角色从技术者转变为管理者，所以我要求自己要与与时俱进，保持不断进步的状态。此外，我意识到需要一个很好的核心团队。在核心团队的建设中，只有不断引入新的人才和资源，才能保持生命力和竞争力，而这是一个需要时间积累的过程。

在创业初衷这点上，我可以自豪地说自己当年最初的想法就是专注Web安全，尽管当时是比较粗的设想，但现在来看当初选择这条道路确实是正确的。更为有趣的是，云安全市场规模的扩大可能超出当初的想象，尤其像现在云计算、物联网所带来的应用安全、数据安全的市场要大于当初纯粹互联网的市场。

**记者：在您看来，国内创业环境适合公司的发展吗？**

范渊：我觉得目前国内创业的大环境是非常良好的，创业园区的支持工作也做得非常到位，因此我们可以集中精力来把自己专注的产品和市场做好。同时，我认为创业也需要国家的其他支持，比如对知识产权的保护，如果这方面做得不够的话，肯定会潜移默化地扼杀一些有创新力的、有真正技术含量的公司。希望在中国良好的创业环境背景下，我们都能够向创业板发展。P





吴朱华  
人云科技创始人

## 做快速满足用户需求的产品

**记者：您开始是在IBM中国研究院工作，后来为何想到要创业呢？**

吴朱华：2007年9月，我进入IBM中国研究院虚拟化小组。在这个小组中，我接触到了虚拟机管理方面的工作，也就是如何在数据中心里通过合理地调配和部署虚拟机的集群来方便管理和节省能源。我们主要试验的虚拟机系统是开源的Xen和VMware ESX，当然在刚开始接触虚拟机管理时，云计算这个概念还没有诞生。同年年底，IBM云计算的传道者Dennis Quan来到IBM中国研究院做云计算相关的演讲。在演讲中，他描绘了云计算的美好蓝图，同时也让我意识到自己当时正在研究的东西就是云计算，而且它非常有潜力。在之后近两年的研究中，随着对云计算认知的不断加深，使得我对云计算的未来越来越充满信心。2009年底，为了进一步参与到云计算大潮中，我离开了IBM研究院，开始了我的创业生涯。

**记者：那您创业的历程怎样？**

吴朱华：其实刚开始并没有什么方向，主要还是深入学习一些优秀的产品和技术，如Google三大利器MapReduce、GFS和BigTable，还有Amazon的Dynamo等。在学习的过程中，我建立了“人云亦云”博客，并写了一些关于云计算的文章。通过这些文章不仅让我有机会分享自己对云计算的理解和认识，而且也加深了对云计算技术的了解。

在研究技术的同时，我从2010年下半年开始着手产品的开发，选择了云数据库和PaaS这两个方向，开发出“YunTable”云数据库和“YunEngine”PaaS系统。在YunTable社区，我结识了几位志同道合的朋友，大家一起筹备创业的事宜和产品的开发。今年初，我们成功拉到了上海张江科技的投资，同时得到了中科院计算所上海分所所长孔华威和弯曲评论首席科学家陈怀临两位师长的支持，因此，我们PeopleYun团队得以在4月中旬成立人云科技。

**记者：为何选择云数据库和PaaS这两个方向？**

吴朱华：现在每个企业所存储的数据量不断攀升，从过去GB级别，发展到现在的TB乃至PB级别，而且其中很大一部分是需要管理和分析的结构化数据，这使得过去基于关系型数据库的解决方案已经无以为继了，所以需要有一个基于分布式技术的云数据库来解决这些问题。

虽然在这方面，业界已经有类似HBase和Hypertable这类的开源产品，但由于它们的整套设计都是基于BigTable的，所以对于Google的需求能很好地满足，但对于大多数用户的需求还无法完美地支持。我们的YunTable核心设计也是延续自BigTable，但在这个经典的基础上进行了创新。再加上我们年底将推出OLAP和OLTP两套全面的解决方案，这些都保证了YunTable无论在伸缩性、性能还是在使用方面，都非常不错。

基于我对云计算的理解，我认为由于PaaS在使用和成本等方面的优势，从长远看来，它将很有可能替代IaaS成为云计算最主流的模式。但无论是Google App Engine还是Salesforce.com都存在一些私有的API，这样将不方便用户进行应用的迁移。同时在反应速度方面，Google App Engine也存在一些不足。为了更好地迎合PaaS这个趋势，我们开发了YunEngine。我们有信心不断地完善它。

**记者：你们团队有何计划和目标？**

吴朱华：我非常欣赏华为这样的国内企业，因为它们做事脚踏实地，而且能快速地满足用户的需求。在初期我们会向它们学习，踏踏实实地把事情做好，帮助用户解决一些非常棘手的云计算问题，并且把公司运行起来。尽可能把公司做强，然后再对用户的需求进行深耕和挖掘，结合最新的技术趋势来进行产品方面的创新，并在3年内通过我们团队自身的积累，争取成为云计算技术方面的佼佼者。P



关注敏捷开发领域的程序员，对Fred George并不陌生，他是有近40年经验的国际敏捷领域大师级专家、咨询师、架构师。2011年3月中旬，他再次来华访问。值此良机，《程序员》杂志采访了Fred George，让我们一起分享他关于敏捷开发的精彩见解。

# 我倡导无政府主义编程

## ——Fred George访谈录

《程序员》：很多人为了编写易变更的代码，在采用敏捷时，抛弃许多习惯用法，由你的经验出发，这样做是否属于重造轮子？

Fred George：这一行为实际是从传统编程转向面向对象编程。我目前的编程方式也有所不同，并且这个新方式与敏捷方式是兼容的。比如我曾经写过大的Java应用程序，那里面平均一个方法只有不到3行的实际代码，平均一个类使用不到25行的实际代码。我也曾经用有1400个类的新系统来替换只有72个Java类的系统。这只是不同风格的编程方式罢了。

因此，如果你打算写大的类和大的方法时，你会发现它们将很难被改变，这也往往会阻碍敏捷程序的进展，让程序员感到沮丧，导致项目失败。如果你写小的类，每个类只做一件事情，并且不允许其他的类插手这一事情，那么程序修改起来就会变得更加容易。任何变动都不会触及其他类，它们将在修改中完好如初。

《程序员》：敏捷开发者可能这样想，“工具不能让你变得敏捷（尽管有所帮助），管理体系也不能让你变得敏捷（也会有所帮助）。敏捷的成功，植根于士气高涨、充分授权的工作者身上”，是否体现了敏捷的实质？

Fred George：对，完全正确。敏捷并不是关于工具或者管理的，而是关于程序员在做他们认为正确的事情。我最近开始探讨一些敏捷中必要的信任转移的话题。传统的思路是，客户提出他们想要的，一大群BA抓住这些要求进行细化，随后PM将其分成小任务，分配给各小组，小组长再进行进一步划分。在此过程中，客户和程序员之间经历了多重分离。客户的零

散需求可以被程序员一一满足，但是客户本人却始终不能与程序员沟通！

敏捷尝试着在客户与程序员之间建立持续的对话。在双方精彩观点双向流动的过程中，信任关系也确立起来了。然后，正如您所说的“士气高涨、充分授权”的程序员就开始理解问题的实质，并为客户提供快速、持续的解决方案。

《程序员》：如能得到准确的数据支持，敏捷实施能够更好地开展下去，请问如何量化敏捷方法的实施？另外，敏捷实践下的程序员工作指标又将如何衡量？

Fred George：我过去常探讨关于如何衡量一名程序员的问题。很多衡量技巧并不是与客户价值相关联的，因而相当复杂。敏捷技巧中的结对编程，使这个问题变得更加复杂。再加上敏捷管理方法，往往会将最困难的问题分配给最好的程序员，将最简单的问题交给新手。这样看来，什么衡量法才能奏效呢？

如果你确实想要知道一名程序员的价值，去问问跟他合作的其他人的意见吧。观察伙伴对待他的态度。最好的程序员就是那种人人都抢着跟他搭配的程序员，反之，人人避而远之的那位就是你应该抛弃的对象了。谁承担了难度最大的卡片并且准时交付任务？谁是其他程序员寻求帮助的对象？通过细致的观察，对一个最好的程序员的判断结果就是显而易见了。尽管去问你的团队吧！

《程序员》：TDD被很多敏捷开发者奉为主壁，但也有很多开发者避之唯恐不及，请问您如何看待之中的差别？此外，测试真的能保证敏捷的实施成功吗？



Fred George: TDD是敏捷的奠基石,尤其对新的敏捷团队来说,更是如此。如果有团队避开了它,那可能是没能真正理解它的价值。

写测试代码首先达到了以下几个目标。第一,它保证了你已做好写代码的准备了。如果你心中没有计划的话,很难写出测试代码来。反之,如果你写不出测试代码来,可能是你做的计划还不够。第二,最好的代码应该是那种设计出来为其他代码所用的。而测试代码则成为此代码的第一使用者,并且通常能带来清晰的界面。第三,测试能告诉你什么时候该停止写代码。程序有时候很容易被写过了头,也就是说,为了解决可能的未知情景,很可能写出来的代码超出实际需求。这不仅会耗费过多时间,还会产生过多冗余代码,也可能会带来更多漏洞,这都是我们不希望看到的。最后,一大堆自动形成的测试能告知你何时破坏了哪些原有代码。

TDD有如此多的优点,为何仍有些团队不愿意采用呢?通常来说,这种团队可能是“没有足够的时间”。殊不知,这种团队会写过多的代码,产生更多难于发现的漏洞,生成繁杂的界面。这种额外的代价是看不见的,相比之下,TDD显然更经济。

对于新团队来说,他们通常对TDD持怀疑态度。但是当他们亲眼看见我使用它在同等时间内写出了比他们多三倍的代码时,他们开始考虑,然后尝试使用,最后,基本再没有抛弃过它。

TDD不能保证成功,但是缺少它却往往能导致失败。

《程序员》:对于未来几年敏捷开发的发展,您希望看到哪些新方向?有何建议?

Fred George:我提倡“无政府主义编程方式”。它支持敏捷方法的所有原则,但是提倡抛弃许多常见的操作实践。我认为它是自然而然的、敏捷和精益方法的进化,虽然有些同事喜欢称它为“后敏捷方式”。它也是Facebook所采用的模式,并取得了成功。简单说来,这个方式就是要求企业为它们的系统设置一些目标,然后在无人监控的情况下,由程序员实施完成所有细节。错误当然不可避免,但是程序

进展的奇妙节奏与当今网络社会的需求相当吻合。他们要做的不是尽力避免错误,而是聚焦在快速发现并改正错误。真正以快速方式轻易解决错误,“快速的失败”远胜过“预防错误”。这当然对客户与程序员之间的信任关系提出了更高的要求。

《程序员》:对于精益等敏捷方法的流行,你持如何的看法呢?这是否是一种新的吸引眼球的管理风尚呢?

我认为“精益”只不过是敏捷的另一个时髦术语。20世纪80年代,我读硕士时就学习了“精益”(当然那时候它的名字还不是这个,而是“Just In Time”,简称JIT)。到了1990年代,我将它应用于敏捷项目的编程中。现在新专家们将这种方式称为“精益”,但它其实还是我一直惯用的敏捷方式。

但是为老的思想贴上新的标签还是很有价值的,它会因此获得更多受众,导致更多的人采用更好的方式。所以看到TDD被重新命名为DDD(领域驱动设计)或者BDD(商业驱动设计),是件很有意思的事情,但无论如何,给新的转变赋予新的名字还是有价值的。

《程序员》:目前中国也出现了很多敏捷“教练”的角色,您对此一趋势如何看待?

Fred George:我不信任敏捷“教练”这一角色,我觉得这种类比是不对的。体育领域中的教练辅导运动员如何表现更出色,但是他们自己不需要身体力行。教练本身都是年纪比较大、反应比较迟钝的。

事实上,程序员们在目睹敏捷做法的过程中获益。顾问需要和他们一起写代码、写测试、部署系统。而许多敏捷教练仅仅告诉你做什么内容,然后坐在一边看着,保证你确实在按照他的要求做。程序员们对此会持怀疑态度,一旦教练离开,他们就马上放弃了这种实施。

敏捷顾问不同于体育中的教练,敏捷的推崇者应当坐下来与团队一起工作,并且身体力行引导团队。程序员不会“太老”或者“太迟钝”;但对敏捷教练,会有“太懒,不愿动手”的说法。P



黄易山

1997年毕业于卡内基-梅隆大学。2001年加入PayPal，曾任高级工程总监。2005-2010年在Facebook领导研发，在公司研发环境的建设上发挥了重要作用。

## Facebook工程管理揭秘之三

# Facebook的内部晋升

文 / 黄易山

Facebook前工程总监黄易山（Yishan Wong）撰写了一系列文章，很好地总结了Facebook卓越研发文化中的宝贵经验。本刊将继续连载这一系列，本文是第三篇。

建设一家健康长久的公司，“从公司内部提拔管理者”是一条广为人知的建议。这条建议也同样适用于规模较小、发展迅速的创业公司。

## 内部晋升的困难

对于超速发展的创业公司来说，秉持内部晋升的方针既非常有必要，同时又非常困难，具体有以下几个原因。

■ 由于公司的超速发展和组织的壮大，因此对于新鲜血液的需求也就非常大。普通员工确实可以从外部招聘，但寻找领导时，想要完全屏蔽外部渠道就变得更加困难。公司主管必须对此保持非常清醒的认识，并且经过深思熟虑后，才能下定决心不从外部招聘领导者。

■ 外部渠道常常被看作是不可思议的完美人选的来源（外面有这么多合适的人选），但实际并非如此。一个成功的管理者需要理解公司文化和公司价值的精髓部分，这通常包括：是什么造就了这家新公司的不可复制的成功以及下一步应该采取哪些步骤。一份令人印象深刻的简历，或是大公司里的同事对他业绩的评价，都不足以证明他能胜任这份工作。

■ 公司内部不一定有足够多的能完全胜任的内部人选。创业公司的主体人员主要由

优秀的普通员工构成，而领导的候选人则一般是他们之中最优秀的。所以从前面两个原因可以看出，若提拔这些人会直接导致他们的业务能力下降，并且也不能保证他们在做管理工作时会像做技术工作时一样优秀。

## 从外部招聘的弊端

尽管内部晋升存在上述困难，但从外部招聘经理或高管并非良策。

■ 招聘经理或高管本身就是种冒险。

任何招聘从根本上来说都是一种赌博。即使进行了细致的面试和筛选工作，你仍然不清楚最终被雇用的人是庸才还是天才。工程师的招聘成功率不会超过80%，经理或高管的招聘成功率则不会超过50%。

这意味着，一名从外部招聘来的经理，有50%的可能会是一个不但不擅长工作，而且会给公司带来致命损害的人。一名糟糕的工程师意味着一场灾难，但通常不是致命的，而糟糕的经理或高管则意味着致命的灾难，公司有可能无法从他所带来的灾难中恢复过来。

你要明白一点，经验丰富的管理者会经常参与招聘，因此他们非常擅长让自己在面试中显得很优秀。

■ 外部招聘的经理很可能会使你放慢公司的发展速度。

这是因为经验丰富的管理者通常来自于大公司，小公司难以培养出很多管理者（因为没有那么多人需要管理，如果有的话，那它就不能被称为小公司了），所以大部分的管理者都来自于大公司。外来的管理者会理解、保持并巩固企业内部文化的概率并不大，他们很可能带来之前任职过的大公司的文化。这会成为减缓公司内部执行力的最强的一股力量，因为他们往往会在时机明显不成熟的时候就引入一些工作流程和方法（而理由通常是，为公司规模的扩大做好准备）。

当公司或工程部门的规模在某一水平之下时（具体地说，是拥有150~250名员工时，这个数目与Dunbar's number（邓巴数）相符，即一个人能够与之维持紧密人际关系的人数上限），企业文化刚刚开始产生，而且起主导作用的仍是每个人的个性和团体的惯例。领导者在企业文化的塑造中所起的作用非常强大，所以一名新领导很容易让整个创业公司将注意力从公司真正的成长方向和市场前景上转移开，从而导致公司失去竞争优势。

不止一家创业公司由于一时疏忽搬起石头砸了自己的脚，它们往往在第一轮或第二轮融资后，就决定要招聘一位职业经理或者副总裁来“帮助团队成长”。但实际情况是，企业规模扩大了，工作流程得以落实，而执行力则大大下降，市场目标变得不明确，若不及时解聘这位经理或高管，公司将无法继续发展下去。

## Facebook的内部晋升经验

因此，成功地坚持内部晋升的方针不仅非常重要，并且也异常困难。这里有一些方法供参考。

■ 寻找那些愿意以普通员工身份加入公司的管理人选。如果公司还是低于一定规模的话，才能出众的技术经理非常适合在这时以普通员工的身份加入，并会快速成长为一名管理者，而且你应该鼓励他们这么做。

这也是筛选出那些曾任职于大公司，但也有能力管理好小公司（或者可以很快领会公司文化，胜任这一工作）的不可多得的管理人才

的好办法。同时，这也是对拥有独特技术才能的技术型领导的一种有效考验，因为他们一次又一次地证明自己能力的信心。拥有在类似创业公司工作的直接经验的人才，通常分为以下两类。

一类是那些曾经自己创立创业公司，但是因为种种原因（比如说收购）而成为大公司的职业经理的人。

另一类是那些曾经在某创业公司规模还比较小（和你现在公司差不多规模）的时期工作过的经理。

如果一名经理没有过在像你这样小规模的公司任职的经历，不要雇用他。因为他们肯定无法融入并运营你的业务，而且还会戴着“凭我在大公司的经验，这件事应该这样做”的有色眼镜看问题。那样的话，你的工程部门、你的公司，都必死无疑。

■ 让少数通过内部渠道晋升的经验丰富的经理对较缺乏经验的经理进行指导，以训练他们的管理能力；如果这个办法行不通，也可以从外面请管理培训师进行指导。一般来说，大部分管理者人选都会展现出自己的管理潜质，但还无法完全胜任这个角色。直到他们完全胜任了自己的角色，对于个人、同事以及整个团队来说才是真正渡过了困难时期。

新任经理应该得到特别的关注，最重要的是让他们明白如何才能适应自己模棱两可的身份，担负起员工管理的重责，并且建立起信心。长期缺乏关注，再加上创业公司超速发展带来的压力，通常会让他们马失前蹄（被降职或自行降职），还会造成团队士气的低落。

经过慎重考虑、权衡利弊后，最终得到的结论是：经过频繁且适当的指导，一名对公司文化有着深刻理解、具备领导潜质且曾对公司发展目标有过贡献的内部人选很有可能顺利成为公司的一名管理者。这样的内部人选远强过一个实际能力、文化倾向和动机都存疑的外部人选。

## 长期影响

一旦组织的规模成功地超过150~200名员工



（邓巴数）这一阶段，公司文化已完整保留并渗透到运营惯例中，也开始自我加强，这时可以逐步地引入外来管理者，让他们直接就任管理职位而无需内部晋升（无论如何，继续通过内部晋升来任命大部分的领导职位仍是个明智的选择），如此可以使管理层的技巧变得成熟。这时，公司文化已经强过任何新任经理的影响，而且会变成一股巨大的力量，排斥那些不能融入这种文化的管理人选，从而得到自我加强。

对于早期加入网景公司（Netscape）和后期才加入的人，Jamie Zawinski有几句著名的描述：早期加入的人是“为了创造一家伟大的公司”，后期加入的人们则是“因为这是一家伟大的公司”，而且在他看来，这一点与这家公司之后的黯淡表现有着重要联系。

后期加入公司的人的动机——特别是在公司取得巨大成功（更“糟”的是，在公司取得巨大的财务上的成功）之后加入的人的动机是非常值得怀疑的——他们有可能是为了钱或安全或稳定，或者三者兼而有之，也就是“因为这是一家伟大的公司”，而不太可能接受公司早期的核心价值。这不是我们希望看到的，而且它会成为损害公司更长远的成功和活力的罪魁祸首。具有这种动机的管理人员被雇佣后，会对公司未来的决策带来很坏的影响。

避免这种情况的方法就是在早期培养足够的领导后备人选，并且最终成功地将他们提拔到领导岗位上来，从而建立一个晋升渠道，晋升渠道中的人选主要是那些早期加入而且“为了创造一家伟大的公司”的人。新加入的员工也可以作为这个渠道的资源，形成一个不断前进的系统。在将他们提升到有影响力的领导岗位上之前，应向他们灌输适当的企业价值观念。P

## 厂商新闻

### MIIT\_CSIP-UC移动互联网应用创新中心成立

3月29日，优视（UC）与工业和信息化部软件与集成电路促进中心（CSIP），携手创建国内首家国家级移动互联网应用创新中心——“MIIT\_CSIP-UC移动互联网应用创新中心”，也是移动互联网行业首个国家级产业公共服务平台与企业共建的行业研究机构。该中心成立后，将会首先针对国内复杂的移动网络环境与终端环境，依托优视的移动互联网软件开发经验，对移动互联网软件质量进行专项研究，为政府指导提高移动互联网软件质量、促进移动互联网行业发展提供建议。

### 英特尔发布至强E7系列处理器

继去年发布至强7500处理器家族之后，4月6日，英特尔推出了32纳米工艺制程的至强E7处理器家族（代号为Westmere-EX）。至强E7可以集成10个支持英特尔超线程技术的内核，比至强7500系列处理器性能提升最多可达40%。

### IBM携手富通东方成立中国首个Tivoli卓越中心

4月12日，IBM公司携手北京富通东方科技有限公司，成立了中国首个Tivoli卓越中心。据介绍，双方将基于ISM（Integrated Service Management）利用各自的技术、方案、专家优势，打造一个包括软件展示中心、培训、支持和服务平台以及专家团队在内的综合资源中心，帮助用户全面了解最新技术与解决方案，依据资深专家建议进行最佳业务实践。

### 诺基亚E7登陆中国

日前，诺基亚召开E7及移动应用解决方案深度体验会，会上展示了诺基亚和微软联盟之后在中国正式发布的第一款高端商务机E7。E7除具有4英寸悦幕、本土化应用、全新Ovi地图支持等亮点外，还加强了对微软Exchange系统的支持，包括邮件、日历任务和联系人的同步等。由微软为其定制的Communicator Mobile 2.0软件提供了包括在线状态、IM即时通信等类似PC上的企业级技术通信体验。



张磊  
风灵创景CEO

# 做可延续的运营平台

记者 / 张雪峰

风灵创景CEO张磊认为一个优秀的操作系统不仅仅是孤零零地与硬件配合，更应该是一个帮助用户提升设备整体性能的运营平台。

记者：请讲述一下您创业之前的经历。

张磊：我在2000年进入IT行业，当时有两个行业热点：一个是电信业务正在蓬勃发展，另一个就是互联网开始崭露头角。华为作为行业内的先锋企业，给予了那时的年轻人十分广阔的空间，所以我加盟了华为。

在华为的业务体系下，基本上都是大兵团作战，一个普通的项目动辄需要几百人的团队，交付周期经常需要1年，甚至2~3年。按照现在的评价标准，这样的工程根本不算业务，应该叫做软件平台。所以它与现在的互联网业务不同，不一定要快速、灵活，只需要去完整地解决用户的终端需求。

经过在华为几年的摸爬滚打，2008年我打算创业，并开始做一些热身和储备。但是，因为年底金融危机的爆发等诸多因素，行业的形势又发生了新的变化，使我应该先到一家大型互联网公司去练练内功。而百度在众多互联网企业中，是一个具有开放体系的公司。在百度我可以接触到互联网的上游和下游，以及各种互联网形态。我觉得这种公司环境非常适合去学习和发展。鉴于这样的原因，我毫不犹豫地加入了百度，做无线领域的业务。

2009年可以说是移动互联网元年。对于整个行业来说，无论是从用户规模、领域投入，还是从对移动互联网的认知度，都达到了一个非同以往的高度。而此时我在百度无线业务部门已经获得了充分的知识积累，我觉得创业的时机已经到来。

记者：您是怎样参与到了李开复先生的创新工场的？创新工场对您的创业给予了哪些帮助？

张磊：我当时有两个创业方向：一个方向是做女性社区；另一个方向是做移动互联网的运营平台。当时，这两个意向都有投资商愿意投资。经过朋友的引荐，我与创新工场进行了简单的交流，大家都觉得“情投意合”，便很快确定了合作关系。

我觉得创新工场是一个非常优秀的投资商，它与我们团队的投资协议非常优惠；它会真心地帮助创业团队快速成长，而不是每天斤斤计较、算计你应该如何做。在这种环境下，我们很快就完成了最初定下的三个目标：融资、组建团队、出产品，我们自己非常满意，投资商也特别满意。

创新工场对所投资团队的帮助主要体现在两个方面。第一，一家新成立的公司比较难得到客户的信任，而创新工场的品牌，还有开复的人品是值得大家信任的，这使得我们在跟客户去谈判的时候，容易赢得他们的信任。同时如果要进行国际化的运作，也比较容易通过投资商获得一定的资源。第二，公司在初创的阶段会遇到法律、财务、政府关系等一系列的问题，创新工场可以很好地帮助我们解决这些问题，使团队能够聚焦产品和服务。

记者：您的团队从最初的十几个人，到后来的九十多人，在管理上这是一个很大的转变。您的团队在管理模式、业务流程上有哪些创新？

张磊：我相信任何一个团队都会有从小到大的复杂过程。最初我们的团队是与创新工场孵化器中的其他研发组一起工作的。后来我们认为，应该作为一家公司去经营我们的业务，产品研发也需要聚焦，于是决定从

孵化器中脱离，开始独立经营，并把团队的全部精力聚焦到点心OS上，这也宣告了点心OS的正式成立。这之后，虽然我们做的事情比以前少了，但是做事的效率高了，团队的凝聚力也提升了。

在公司规模逐渐扩大的过程中，我觉得领导者的能力是非常重要的，例如选择合适的人才、协调团队成员的关系、统筹规划产品的流程等。当然，互联网公司的管理模式与传统公司是不同的。传统的管理模式，注重流程和组织结构的清晰，以及管理的有效性。而互联网公司的管理模式更加注重发挥个人的能动性。我们希望团队中的每个人，不仅要聪明能干，还要有战略性和系统性的思维能力。因为，互联网行业讲求的是创新、快速，它的竞争要比传统行业更加残酷，所以更需要把人的能力发挥出来。现在我们公司的管理，就是强化人的能力和作用，弱化业务的流程。当然，流程在企业中的作用也不能被忽视。我们的员工最近经常自发地讨论如何保证流程执行的效率和质量。我们并不会很僵化地要求员工来如何执行某些要求，而是让员工自己去管理业务流程，这也许是互联网精神的具体体现吧！

有很多人认为，强调了个人能力，就会牺牲团队的合作性。但我们认为如果一个人的能力很强、为人处世的水平很高，那么他一定会有非常好的办法去解决流程中的问题和工作中人与人之间的矛盾，这样反而会形成合力。

**记者：**您在带领团队开发点心OS的过程中，技术方面有哪些创新？由此产生了怎样的体会？

张磊：在宏观上，操作系统是一个运营平台，而不是一个孤零零的与硬件配合的软件，所以在设计过程中特别强调运营性。我们在设计点心OS的过程中，特别加强了云端服务、交互性的设计，着力将点心OS打造成一个优秀的移动运营平台。对于操作系统的开发，不仅要做好眼前的产品，还要深入考虑到产品的延续性，做成一个容易进一步开发的产品，甚至使它成为产业。

在微观上，操作系统是最靠近硬件的软件，所以对于提升设备的整体性能起着至关重要的作用。我们在这方面做了很多工作。比如为了让手机更省电，我们在驱动软件层面对于算法进行了优化，节省了大约10%的电量；而在系统软件层面上，我们会自动监测你不经常使用哪些服务，比如像GPS经常打开会耗电，我们的系统则可将这项服务固化成省电模式。

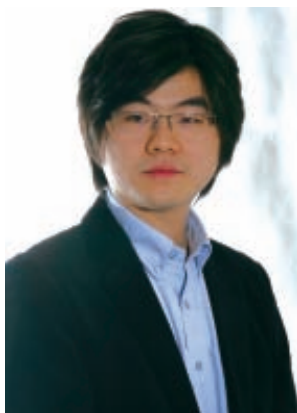
**记者：**我们看到点心OS的界面非常华丽，去掉了一些Android原有的UI元素，并根据用户体验做了许多改进。点心OS的这些改进具体体现在哪些方面？

张磊：我在微博上说过点心在做的三件事：“打个比方，Android是个美人胚子，点心的一部分工作就是给她强身健体，这部分就是优化OS的本身技术体系，做好硬件适配；第二部分就是给她整整容、上上妆，这个部分就是UI体系；第三部分就是教她知书达理，提高内涵，这个部分就是点心的云服务。三个部分做到了，才是一个好系统。”

就UI来说，我们首先设计了一套兼容性很好的UI框架，这个框架可以保证我们每年至少出两套大的交互模式，每一套交互模式中又有很多的主题风格，这样就保证了给UI设计师充分的想象和发挥空间。去年，我们做了一整套色彩偏重、Widget可以撑满屏幕、图形效果酷炫的UI。今年3月份我们上市的手机又支持了一套3D龙卷风效果、同时支持键盘操作的UI模型。今年6月份的时候，我们将会出一套清爽、透明、简洁，专门面向白领用户的UI。

另外，我们与只做UI的团队不同。我们要考虑的不仅仅是把界面做得酷炫，而且还要让用户觉得我们的产品好用，这就涉及到性能的优化、以及提供更多优秀的应用和服务等问题。P





黄晶  
人人网技术总监

# 用熟悉的技术做产品

记者 / 张雪峰

人人网技术总监黄晶认为一个软件系统是不是健康、能不能持续发展，与整个系统架构的设计有重要关系。而从性能上来说，不同语言之间的差别并不是那么重要，所以在技术选型中，最好是应用自己和团队熟悉的技术。

**记者：**您曾经在天人互动游戏公司和空中网任职过，后来为什么选择加入千橡了？

黄晶：在我刚加入千橡的时候，公司条件还是比较艰苦的，作为一家创业公司，没有多少技术积累，资金也不多，办公地点是一幢商住两用的楼。我之所以选择加盟千橡，首先，是对互联网社区的前景充满信心；其次，我相信陈一舟的战略眼光和商业运作能力，觉得他是一个可以做大事的人。

**记者：**千橡收购校内网之后，您对校内网的技术架构做了哪些调整？

黄晶：校内网被千橡收购时，还是一个创业团队，人数较少，所以架构设计也很简单，典型的Java+MySQL。但随着访问量的逐渐增加，数据库和中间层缓存的结构已无法适应当时的发展速度。所以，我们对数据库、中间层、前端Java框架都做了彻底的重构。

在产品上，过去校内网是把个人主页作为了解好友和沟通的最主要方式，就相当于为每个用户提供了一个表达个性的涂鸦板，这是早期校内网的核心；后来，我们将“新鲜事”这个功能加了进去，以“新鲜事”作为最主要的获知好友动态的方式，实际上“新鲜事”成为了信息流动的核心。

**记者：**现在有很多公司都喜欢用PHP来做SNS，千橡为什么选择用Java来做？

黄晶：使用Java做SNS，主要原因就是千橡从最开始做网站就一直在用Java，所以有非常多的积累，从语言特性、虚拟机优化到开发框架和部署的体系都比较完善。Java有强大的Sun公司和开发者社群在支持着它，发展非常稳健。在广大的开源社区支持下，积累了很多开源项目和组件包，无论你想实

现什么，都可以找到需要的东西。对于整个产品来说，语言不是很关键的因素。Java语言面向对象特性极强，可以把整个体系组织的层次模块清晰，结构优美。但是它的代码量、冗余度可能会高一些，不那么简洁。

**记者：**人人网作为一家国内著名的SNS，您的技术团队都采取了哪些措施来保证海量数据的安全性？

黄晶：首先，我们已经花了很多的精力与资金来保障数据的安全性和服务的高可用性。任何服务器都会有硬件故障，我们要保证任何一台甚至是几台服务器发生硬件故障的时候，数据不会丢失，整个系统仍然能够正常运转。在这个方面要做的事情非常多：我们设计了一定的存储冗余度，并通过系统机制来实现线上热备份以便在故障发生时及时替换。通过以上手段，我们已经解决了很多问题，可以保证在任意一台服务器宕机之后不会对数据安全和用户的访问造成很大影响。

其次，我们正在进行基础设施即数据中心的扩充。目前数据中心是非常紧缺的资源，要想找到一个规模和质量都适合的数据中心非常困难。随着我们业务的不断扩大，需要不停地加入新的服务器，单个数据中心会趋于饱和。每饱和一个数据中心，我们就需要再新建一个数据中心。但新建的数据中心如何与之前的实现分布式架构下数据的实时同步，如何制定广域网负载均衡的方案以达到最佳访问速度，这又是一个很难解决的问题。而且，为了应对地震、火灾等自然灾害，应该把不同的数据中心建在不同的城市，导致异地数据同步难度加大，这是我们目前正在解决的问题。P

# 软件研发中的冲突及解决之道



时永安

深圳易方数码科技新技术研究主管，微软 MVP

软件项目在研发过程中牵涉到很多利益相关方，这些相关方因为关注角度的不同，会产生很多矛盾冲突。这些冲突，轻则打击士气，拖延项目的进度，重则使项目无法正常进行。在我这些年的软件项目管理工作中，遇到过各种各样的冲突，其中最常见的有：项目开发周期的冲突和团队内部人际关系的冲突。

软件项目的研发周期，本来是应该根据项目工作量和开发人员情况来估算的。但现实中，往往会受到市场部门以及公司高层的干涉。他们从产品销售的角度考虑，希望软件产品越早发布越好，在他们眼里软件开发弹性极大，只要给的压力足够大，就可以成功地将开发周期缩短。作为开发人员，我很清楚如果按照他们的要求随意确定开发周期，就意味着无休止的加班、低沉的士气以及进度的一再拖延。这就有了冲突，因为双方都觉得自己有道理而不愿让步。出现这种情况时，互相妥协是唯一的办法。项目主管可以通过安排适量的加班，削减或者推迟部分功能的办法来做出一定的让步，但这种妥协一定要有底线，过度的妥协会导致一个无法达成的开发进度计划，对项目造成极大的伤害。

如果说上述冲突是“外患”的话，那项目团队内部的人际冲突就是“内忧”。软件开发人员往往表面看似低调，其实内心骄傲，他们对自己的智力充满了自信，最无法容忍的就是自己的工作成果被否定。进行Code Review及Bug责任人确定时，最容易引起内部冲突。处理这种冲突有赖于项目主管的管理技巧以及公平的处事原则，同时把“对事不对人”的工作态度灌输给项目中的每个成员。冲突发生时，既要坚持原则，有理有据地作出分析，也要注意照顾双方的情绪，多做安抚工作。

总之，软件研发过程中的冲突无法避免，但也不是洪水猛兽，只要积极应对，就可以将负面影响降到最低。



崔康

IBM中国研发中心软件工程师

软件研发团队会不可避免地遇到各种主观或客观的冲突，如何较好地解决矛盾是团队成熟与否的重要标志。我选择了两个典型案例，希望对读者有所启发。

## 时区/地域冲突

这是一种客观冲突。全球性的研发团队包含了来自不同时区和地域的工程师，在这种情况下，应该采用哪些“协作”方法来降低冲突带来的负面影响？处理这种冲突的关键在于提高沟通的质量，一方面要尽量不影响各地工程师的正常作息，另一方面要通过合适的安排来最大限度地利用时间差。我有以下建议。

1. 团队例会时间由各个地区的工程师共同商议确定，选择一个大家都“醒着”的时间段，如果困难，则采用定期轮换的方式让大家公平地“享受”熬夜。开会采用视频或者电话形式，不鼓励使用文字交流，书面会议纪要由团队成员轮流记录。会议之前确定议题方案、备选方案、决策规则等，提前发给团队成员，开会时直奔主题，不延时。会议纪要必须包含下一步的行动计划，具体到每个人，得到大家认可后立即散会。

2. 工作安排上，尽量利用时间差。换句话说，在某时区白天工作的成员要把当天的结果通过邮件等书面形式及时反馈给另一时区（稍后上班）的同事，形成良性循环，以便大家都能够在自己上班时看到前一时区同事的反馈并据此开展工作，然后在自己下班时将进展和问题反映给后面时区的同事。

新旧“势力”冲突

这是一种主观冲突。随着技术的不断发展，研发团队经常尝试引入一些技术或者方法，在这个过程中，新旧势力会发生冲突。当事一方是项目经理或者架构师，他们对新技术充满信心，并想尽快实践；另一方是团队普通成员，他们已经习惯了手头的老技术，应用自如，对新技术的实际作用存在疑问，不愿尝试。如果不能很好地解决，必然导致双方互相埋怨，新技术推广艰难。分析其中原因，除了必要的沟通，还需务实的计划和实践。我有以下建议。

1. 引入新技术和方法前，通过学习会议等形式让所有成员对新技术有充分的了解，但不要强制他们认可并马上实践。
2. 制订循序渐进的增量引入计划，不要立即整个抛弃老技术，请团队成员一点一滴地试用新技术，逐步习惯甚至喜欢上它。
3. 在技术过渡阶段，团队负责人和架构师等要实践其中，不能置身事外指指点点，及时听取团队各个成员的反馈，并作出相应调整。
4. 定期召开团队会议，让成员根据自己的实践结果对新旧技术做评价对比，培养认可新技术的氛围。时机成熟后，大规模采用新技术。
5. 新技术引入后期阶段，团队管理层对成员表现进行肯定和奖励。



张小明  
趋势科技测试工程师

有人的地方就有冲突和斗争，不管你在，还是不在，只要有人在，冲突就在那里，不多不少。作为一名普通的软件工程师，其实更愿意称自己是一枚程序员。我在国企、私企、外企都写过程序，编译过、运行过、调试过，可谓撒过汗、通过宵也掉过肉，在软件研发过程中自然也经历过大大小小不同类型的冲突。不同的企业有不同的文化（没有文化也是一种文化）：国企的冲突往往被套上了制度的枷锁，私企的冲突往往和大当家的为人处世和经营理念有关，而外企的冲突往往不分尊卑，带有天马行空、各成一派的自由风格。

我曾经在金融软件项目中做研发，该项目的开发模式是与客户（某大型国有银行）的开发部门共同合作，两拨人在一起办公，我们有一个项目经理，银行方面也有一个项目经理。虽然研发的主导力量在我们这边，但从理论上讲，人家是客户，我们是上门服务。冲突缘起于某次项目赶进度，对方项目经理要求加班，而我们有些成员不太情愿加班。这时对方经理要求自愿加班，而我们的项目经理带头拒绝加班，根本不给客户情面，于是导致客户非常愤怒。接着冲突升级，对方经理提出要致电我们老板要求当家的出面强制加班，结果冲突再次升级，老板的出面还是没能让我们项目经理接受加班。后来团队里抵制加班的相关同事被通报批评，并且被强制在客户面前书面道歉，而这件事的直接后果是这个项目组中的我们公司的研发人员全部跳槽。这是一次没有善终的冲突。

作为一名普通工程师，我认为，既然从事这个行业，就应该端正自己的职业态度，你是为公司打工，更是为自己攒人品。不管当下的待遇或境遇有多差，我们可以选择其他方式解决或者高贵地离开，至少不应该把怨气发到客户的身上，但同时也必须从制度上杜绝以强凌弱的现象发生，比如违约金这种东西。

上面举的例子比较偏激，而且我恰恰认为冲突不应该刻意去避免，优秀的团队文化应该鼓励适当的冲突，并进行引导，然后化解，而不是压抑住冲突（压抑久了就可能会导致上面的“杯具”再次发生）。作为一名程序员，我坚持认为良性的碰撞和冲突能产生火花和灵感，甚至是解决方案，但实现这些需要思考的前提是：如何构建和谐、优秀的团队氛围，从工程师的角度来看，我认为就是端正自己的职业态度。





**Marty Cagan**

过去20年，Marty Cagan作为负责定义和开发产品的高级经理人为多家一流企业工作过，包括惠普、网景通信、美国在线、eBay。他亲历了个人电脑、互联网、电子商务的起落沉浮，致力于通过写作、演讲、培训帮助客户打造富有创意的产品。为此，他撰写了《Inspired: How to Create Products Customers Love》一书，创办了硅谷产品集团公司（SVPG）。在此之前，他的最后一份工作是担任eBay产品管理及设计高级副总裁，负责规划全球电子商务网站的产品和服务。

# 产品管理与软件开发的关系

文 / Marty Cagan 译 / 欧坤，孙洋

Marty Cagan是享有世界声誉的产品管理专家，曾经担任网景副总裁、eBay产品管理及设计高级副总裁。本文是他回顾自己二十多年来从事软件产品管理工作的总结和经验分享，谈到了产品管理与软件开发的关系，以及软件开发人员如何转型做产品管理。

## 产品管理与软件开发的关系

如果说成功的产品是真实用户需求与现阶段可行性方案的结合，那么产品经理与开发团队之间（合作）关系的重要性不言而喻了。

产品经理负责定义产品方案；开发团队最了解哪些产品设计是可行的，他们负责产品的开发与实现。作为产品经理，你很快能体会到，只有与开发团队融洽合作，才有可能开发出合格的产品，否则等待你的将是一段漫长难挨的日子。

形成合作关系的关键是双方承认彼此平等——任何一方不从属于另一方。产品经理负责定义正确的产品，开发团队负责正确地开发产品，双方相互依赖。你要求开发团队完成任务，必须先取得他们的认可，确信为了达到产品质量标准必须这么做；开发团队也要留给你足够的空间设计有价值、可用的产品。

产品管理和软件开发相互促进，开发人员可以帮助产品经理完善产品定义，别忘了他们最清楚你的产品设计是否可行。

开发人员帮助产品经理完善产品定义的方式有如下三种。

■ 让开发人员直接面对用户和顾客，体会用户的困惑和疑虑，了解问题的严重性，好点子常常会随之而来，譬如，可以邀请一名开发人员参加产品原型测试。

■ 向开发人员了解最新的技术发展动向，讨论哪些新技术可以用到产品里。开展“头脑风暴”，看看目前已实现的技术或即将实现的技术能不能解决手头的问题。

■ 让开发人员（或主程序员）在探索（定义）产品的初期阶段参与评估产品设计，协助策划方案。产品经理常犯一类错误，即完成产品定义后，扔给开发团队便置之不理。这样做只会贻误协调需求与可行性的最佳时机，等到发现问题时，为时已晚。

同样，产品经理也可以协助开发人员的工作，方式如下。

■ 产品经理关注产品的基本要求（核心功能）。产品经理应该意识到，自己定义的不是最终产品，而是满足基本要求的产品雏形。只有这样，产品管理与软件开发之间才能形成良好的互动。

■ 一旦产品进入开发阶段，要尽可能避免修改产品的需求和规划。虽然有些事情超出你的控制范围，变动是不可避免的，开发人员也能理解，但是千万不要在此时尝试突发奇想的点子。

■ 产品开发阶段难免会产生诸多问题，比如，用例丢失，或者用例设计考虑不周全等。这很正常，哪怕最优秀的产品团队也避免不了。产品经理应该迅速采取行动，在维持产品核心功能、尽量避免修改的原则上，给出解决方案。

我经常鼓励出色的开发人员尝试产品

管理工作。我告诉他们，如果产品没有市场价值，无论开发团队多么优秀也无济于事。很多优秀的产品是程序员抓住用户需求，自己创业研发出来的。放宽眼界不仅仅有利于开发人员自己的职业发展，也有利于产品、顾客和公司。

## 如何与异地的开发人员沟通？

如今产品经理与开发团队各处一方的情况很常见。除了印度软件外包业务，大型公司的分支机构之间，以及公司与被收购的子公司之间，都可能出现这种情况。

如果开发团队不在你身边，沟通和执行的难度将进一步放大。异地开发出现的问题常常导致团队士气低落，有人甚至公开质疑异地开发能否真正节约成本。

提高与异地开发团队之间的沟通效率，我有三条建议。

- 开发团队距离越远，语言、文化、时差带来的沟通难度越大，确定完善的产品说明文档就越重要。如果产品经理不确定开发什么样的产品（或者反复改变想法），异地开发团队只能疲于奔命，白费力气。这简直就是一场灾难，丝毫不亚于医生开错方子。

如果你与开发团队相隔很远，无论是沟通产品文档的内容，还是讨论修改产品设计，一定要借助高保真原型进行交流。阅读文档毕竟不轻松，如果文档是非母语的，或者阐述不清楚，那沟通效率就更低了。

- 本地的开发团队很容易发现和解决各种冲突（比如，两个管理者给出相互抵触的指导意见）。异地开发团队则会发生很多意想不到的情况，往往过了好几个月才发现问题。这是因为异地开发人员不得不揣摩各种不同的意见（但经常揣摩错）。因此，必须有人在本地负责与异地团队的协调工作。这并不是说所有沟通工作都必须经过此人，而是应该明确异地开发团队只接受他的命令。这项工作可以由本地的项目经理、资深开发人员或者其他主管担任。

- 如今商业沟通手段很丰富，除了电子邮件和即时消息外，还有视频会议可供选择，此

外，语音电话业务（VoIP）大大降低了国际长途通话费用支出。尽管如此，当面交流的优势依然不可替代。每个季度，产品经理至少应该前往异地与开发团队见一次面，与软件架构师、管理者当面交流。面对面交流有助于改善（合作）关系，提高沟通效率。此外，交换人员也是一种有效的沟通方式，可以让主程序员来本地与产品经理共同工作一段时间，或者让产品经理到异地工作一段时间。

按照我介绍的方法与优秀的开发团队合作是一种特别的享受。如果是与印度外包团队合作，那么由于时差的原因这种合作会让人觉得更加惬意。每天早晨上班时，对方的项目进展已经摆在面前。你可以用白天（对方的夜晚）检查、测试代码，反馈信息，显著缩短项目的循环周期。

---

开发人员转型做产品管理有其无与伦比的优势——对产品可行性的敏锐嗅觉。如果他们对用户行为进行深入分析，学习一些和产品管理有关的技巧，就能成长为出色的产品经理，打造出用户喜爱的产品。

---

请注意，如果是在异地开展与产品原型相关的工作，由于循环周期非常短（一天迭代好几次），你必须随时准备处理对方的问题，投入更多的精力。

解决异地开发问题的另一种方式是在异地聘请产品团队。这种趋势正在兴起，我相信这种模式会被更多的公司采用。你大可不必为此担心。我们曾经用了10年时间在异地培养专业的开发和测试人员，培养专业的产品经理和设计师很可能还要再花10年时间。

## 程序员想重写代码？

产品经理最担心听到开发人员这样抱怨：“不能再增加功能了！我们得停下来重写代码。代码库一团糟，就像纸糊的老虎，根本应付不了持续增加的用户。我们维护不下去了！”

这一幕在很多公司上演过，现在依然在不

断重演。1999年eBay遭遇这一幕时，公司濒临崩溃的情形超出所有人想象。Friendster几年前也发生过这种情况，当时他们正在向MySpace开放社交网络用户。网景与微软展开浏览器大战时，也发生过类似的事情，最后的结果大家都知道。事实上，没几家公司能渡过难关。

一旦公司陷入这种困境，开发团队往往沦为替罪羊。经验告诉我，这类问题通常是由产品管理失误引发的。因为产品经理一直迫使开发团队满负荷工作，开发尽可能多的功能。所有软件架构都存在功能极限，如果忽视产品的基础技术构架，一旦系统越过崩溃的临界点，就会造成无法挽回的结果。

在重写代码的过程中，用户无法看到产品的任何改进。你可能认为重写代码至多也就几个月，但是实际花费的时间无一例外要多得多。你只能坐在一旁，眼睁睁看着用户投奔竞争对手，而这个时候，竞争对手恰恰在不断地改进产品。

如果你还没有遇到这样的情形，未雨绸缪很有必要——你需要预留一定的研发技术能力，在eBay被称为余量（headroom）。很多因产品迅速膨胀产生的问题都与扩展规模有关，余量意味着避免触及技术能力的上限，为用户数量的增长预留空间，为事务增长预留空间，为新增功能预留空间，保证产品的基础技术构架能够满足团队的要求。

与开发团队合作应该遵循以下原则：在产品管理上为开发团队预留20%的自主时间，让他们自由支配。开发团队可以利用这些时间重写代码、完善架构、重构代码库中有缺陷的部分，或者更换数据库管理系统，提高系统性能，避免“我们需要停下来重写代码”的情形发生。

如果你的糟糕处境已经初现端倪，应该拿出至少20%的资源进行调整，而我担心有些团队连20%都不愿意拿出来。

如果你已经身陷重写的困境，说明公司危在旦夕，这里给出一点建议供你参考。

第一步，针对开发团队确定的产品修改目标并制订切实可行的计划和时间表。通常，有经验的开发团队估计的开发时间八九不离十，

但是重写代码是个例外，因为多数团队都没有重写代码的实际经验，估计往往过于乐观。你必须审时度势，仔细检查每处细节，确保计划切实可行。

第二步，只要有可能，最好把重写目标分成几大块，实现递增修改，让用户感受到产品的改进，哪怕会因此把9个月的工作时间延长至2年，也一定要采用这种方式。重写代码时，保证让用户看到功能的改进——即使会占用少则25%，多则50%的开发资源——对保持产品（尤其是互联网产品）的市场占有率至关重要。

第三步，由于开发用户可见功能的资源有限，必须谨慎选择正确的产品特性，并确保产品定义的正确性。

eBay起死回生后，我们发誓绝不重蹈覆辙，马上开始新一轮的代码重写，把危机甩在身后。事实上，由于发展迅速，eBay已经重写了三次代码，最后一次采用了完全不同的编程语言和架构。开发团队花了几年时间，大规模地改写了几百万行代码，同时在不影响用户群的情况下，开发了大量新功能。这是我知道的最惊心动魄的中途重建网站的故事。

临渴掘井不如未雨绸缪，为防患于未然，别忘了预留20%的余量。如果你从未与开发团队谈过这件事，今天去找他们吧。

## 软件开发人员如何转型做产品管理？

我与开发人员接触，发现他们很关心这样一个问题：如何从软件开发向产品管理转型？

开发人员希望向产品管理转型，有时是因为参与探索（定义）产品后，尝到了影响产品决策的甜头，不再满足于只做编程的工作。有时是因为对现有产品很失望，他们认识到如果产品没有价值，开发团队再优秀也无济于事。

我认识的很多优秀的产品经理都是开发工程师出身。接下来，我将探讨从软件开发转型到产品管理时可能遇到的问题和挑战。

开发人员转型做产品管理有其无与伦比的优势——对产品可行性的敏锐嗅觉。如果他们对用户行为进行深入分析，学习一些和产品管理有关的技巧，就能成长为出色的产品经理，



打造出用户喜爱的产品。

转型的第一步，是清楚地意识到自己和目标客户是截然不同的。花一些时间和真正的客户交往，就会很容易意识到这一点。不要想当然地认为，只要我喜欢这个产品，知道如何操作，用户也一定会喜欢这个产品，知道如何操作。

第二，学会“移情”（empathy），懂得站在用户的角度思考问题。其实，用户并非一无所知的菜鸟，只是他们的工作和擅长的领域与你不同罢了。要做到换位思考，最简单的方法是花时间与用户做面对面的沟通。注意，这并不意味着用户能提出真正的产品需求；挖掘产品需求是产品管理的任务。

第三，转变思想。作为开发工程师，你的任务是优化开发流程和效率。但作为产品经理，你的工作则是定义产品、优化用户体验，打造出用户喜爱的产品。这一点看似容易，只有当你面临一个两难抉择，比如项目发布时间与用户体验出现冲突的时候，你才能体会其中的困难。

第四，保持谦逊的品格。向用户展示产品时，大多数人的反应可能会与你的预期背道而驰，这时谦逊就显得格外重要。仔细倾听来自用户的声音，日复一日，你的理解力会得到极大提升。但前提条件是必须拥有开放的心态来面对用户的批评。

第五，改变讨论风格。很多互联网企业中，工程师都喜欢围绕着某些决策争论不休，激情四射。可多数情况下，这些产品的技术决策有明确的判断标准，比如运行速度更快、规模更合适、容错度更高、扩展性更好等等。而产品定义和用户体验的决策中并不存在这样的标准。这时候就需要你改变以前的讨论风格，提高说服和辩论的技巧，让他人接受你的观点。

最后，处理好和原部门的关系。成为产品经理后，你和开发部门的关系会变得很难处理。他们会变得异常敏感且难以沟通，会采取各种各样的方式来挑战你，质疑你的技术能力，不会轻易作出承诺。你需要学会放手，让工程团队做好他们的本职工作，并参与到产品开发的流程中来。产品管理的工作已经够让人头大了，相关的技术决策就放手让他们来处理吧。

我强烈建议公司建立畅通的渠道为开发人员的转型提供便利，一定会培养出许多杰出的产品经理。即使转型不成功，开发人员还是决定回去编程，但产品管理的观念也为他们树立了“科技以人为本”的思想，对他们今后的工作是极其有益的。P



本文节选自华中科技大学出版社《启示录：打造用户喜爱的产品》一书和作者的博客。该书从人员、流程、产品三个角度介绍了现代软件（互联网）产品管理的实践经验和理念。特此感谢华中科技大学出版社与Marty Cagan先生授权。

## 中科院IT工程硕士研究生 2011年招生信息

以“精技术、懂管理、会做人”为目标，专注于IT领域高层次复合型人才培养；以创新型的“研究生培养与IT人职业发展”相结合的教育体系，为学生发展注入动力之源。

中科院2011年招生领域：

- 软件工程
- 电子与通信工程
- 计算机技术
- 控制工程
- 集成电路工程

■ 软件科技理学硕士：中科院与香港理工大学联合培养，获得香港理工大学硕士学位证书，可获得国际通用GPA记录，留学申请更有竞争力！

招生对象：

大学本科毕业的IT在职人员或应届毕业生，软件科技理学硕士可招收具有三年软件领域工作经验的专科生。

学习方式：

在职学习或脱产学习。

报名时间：

报名正在进行，详情参见我院网站招生信息

<http://ccce.gucas.ac.cn/index.php?id=373>



咨询详情：

中国科学院研究生院计算机与通信工程学院

联系电话：010-88256568/6547/9429

电子邮件：ccce@gucas.ac.cn

学院网站：Http://ccce.gucas.ac.cn

联系地址：中国北京玉泉路19号甲

邮政编码：100049

# 没有永恒的产品经理

记者 / 董世晓

曾担任百度贴吧首任产品经理并主导完成过多款百度社区产品设计的李明远，现任UC产品副总裁，率领独立产品团队打造移动社交平台UC乐园。从百度到UC，在互联网产品的战略规划、用户体验、运营等方面的积累和经验，让他对产品经理职业有独到的见解。

**记者：您怎样看待产品经理职业？**

李明远：现在基本上每个公司都意识到了产品管理的重要性，也设置了产品经理岗位，但合格的产品经理非常稀少。很多人可以称之为是产品团队的一员，但从事产品工作与最终成为产品经理之间还存在很大的鸿沟。我的观点是，只有极少数的人最后能够变成产品经理，尽管很多人会一辈子从事与产品相关的工作，比如项目管理、需求分析、数据挖掘等。某些人在某一个领域非常擅长，在产品团队里面是非常优秀的产品工作者，扮演着很重要的角色，但这距离成为一个产品经理还差得比较远，只有很少人能够迈过这条鸿沟。

**记者：要想成为一名优秀的产品经理，您认为应该从哪些方面进行努力？**

李明远：成为一名产品经理需要很多优秀的特质，但不是说具备哪些特质就可以成为一名优秀的产品经理。产品经理有彼此不同的特质。当然，整个群体也有一些共通性。

**尊重用户：**每个人都把尊重用户挂在嘴边，但真正做到的人少之又少。太多的产品经理是在替用户做选择，已经不再是尊重用户，而是以自我为中心。

**反思能力：**产品经理最忌讳的就是认为自己曾是一款优秀产品的负责人，就一定是一名优秀的产品经理，并想当然地认为做任何产品都没问题。实际上，决定能否成为某款产品的产品经理在于你与产品之间的兴趣、感知、互动以及深入的幅度。因此，一名优秀的产

品经理不一定在哪儿都是优秀的产品经理，反思很重要。要做到非常清醒地回顾先前的产品之所以优秀的原因，并且要牢牢记住：没有永恒的产品经理。

**知识面：**做产品就像做人一样，必须具备一些最基本的知识。其实上面提到的尊重用户，也是知识面的体现。尊重用户更多的是从感性的角度，而这里所说的知识面更多是从理性的角度。不同的业务需要不同的知识，产品经理的知识面要足够宽阔，只有具备更好的学习能力，才能少犯错误、少走弯路。

**记者：产品经理职业源于国外，您认为国外有哪些好的经验值得学习？**

李明远：国外团队执行力强，分工细化，每个工种都钻得很深。而中国很多工作的粒度还是比较粗、扁平化，团队和跨职能的事情往往特别依赖于某一个人。这就带来一个问题：因为这个人某个领域特别强势，那么他就更倾向于发挥自己强势的地方，而他弱势的一面就会做得比较欠缺。

而国外的分工则非常细，在各个关键领域都力求做到极致，很值得我们学习。举个例子，Groupon在创立早期雇了相当数量的地方报纸评论员和写手为自己做宣传。因为这些人非常犀利，有的幽默、有的尖刻，各具特色，写的内容可读性强，容易被地方大众接受。这对Groupon的帮助非常大，因为团购是冲动消费，用户往往就因为这个照片拍得很好，介绍写得很赞，就会被打动，继而忙不迭地参加团购。



李明远认为移动社交是一个趋势

**记者：在百度和UC您都是负责社区产品，感受有何不同？**

李明远：人和信息之间有几个维度——人找信息、人找人、信息找信息、信息找人。百度作为一家搜索公司，其社区产品解决的是人找信息的维度，使搜索信息更方便。而UC乐园定位的是社交平台而不是社区，是看到了移动社交的需求，解决的是人找人的维度。

与PC端应用相比，移动社交有一个先天优势，那就是基于真实身份，可以通过娱乐、沟通、LBS，将人联系起来。因此，移动社交应用具有非凡的吸引力，也会吸引更多的开发者投身于此。而为了使开发更加方便，我们也推出了自己的开放平台，帮助开发者解决终端、平台、网络适配等方面的问题，使他们专注于开发工作。P



主持人：冯大辉

现任丁香园（<http://www.dxy.cn>）网站CTO。曾历任支付宝架构师、数据库团队负责人等职。

# 架构师接龙

## 孙立 VS. 孙朝晖

**孙立：**你是在架构层面，提高开发人员开发效率的？比如通过合理的分层，不同层安排不同能力的开发人员。

**孙朝晖：**首先孙立老师已经谈到了这个问题的两个核心，第一是合理的分层，第二是让不同能力层次的队伍有机组合。

■ 对于分层，具体到我们的技术体系，可以清晰地分成四个层次，对应四个技术层次，分别是：前端（JavaScript开发）、Web应用（PHP开发）、中间件（Java开发）和通信与管理基础（C开发）。各层有独立的团队，开发人员专注于本层次的技术发展，各层次的开发团队Leader每日进行晨会交流开发进度，每周例会进行技术整合研讨。对于较完整的功能模块，设置有“技术方案评审会”，各团队专家参加，通过对各层技术特点的分析，综合考虑方案的可行性。

■ 对于组织结构保证，每个团队都会有1~2名技术专家，1名团队Leader，若干开发人员。每个团队的技术专家负责开发基础框架，比如我们PHP应用的MVC框架、Java的Service Framework、前端JavaScript的基础类库，对于不同层次的开发人员根据能力安排角色，决定担任部件级别开发还是组合功能级别开发，保证每个团队都有不同层次的人员合理组合，每个开发人员都保持技术上升通道。这样会有有效地提高开发效率，同时每个技术人员都有清晰的当前定位和未来定位，有利于技术团队的成长。

**孙立：**要变更数据库结构，有什么经验可以分享？

	<p>提问嘉宾</p> <p>孙立</p>		<p>回答嘉宾</p> <p>孙朝晖</p>
<p>去哪儿网高级系统架构师，曾就职于凤凰网、酷6和搜狐。对分布式搜索引擎开发、大数据量网站系统架构优化、运维监控等有丰富的经验。开源项目phplock和phpbuffer的作者，近期开发了NoSQL数据库存储INetDB。</p>		<p>飞信互联网产品首席架构师。负责飞信SNS业务相关产品，空间、开放平台的总体架构设计与飞信开发团队技术社区建设。2010年加入北京新媒传信科技有限公司。曾历任微软(中国)高级顾问、高级项目经理。</p>	

**孙朝晖：**这个问题是互联网行业的通用难题，数据量大，数据结构变更的代价太大，对于这个问题，我想从四个方面进行介绍。

■ 在技术方案评审阶段，充分考虑数据存储方式的合理性。不是每种数据都适合采用数据库存储，有些数据适合采用JSON，或者采用缓存中的List对象存储，而不适合分解到数据单元，这样的数据就没有必要设计成为关系型结构。

■ 在数据结构设计过程中，保留适当的冗余字段，尤其是预估有动态变更的数据结构，而且状态标志尽量组合成为数据位结构，即典型的Mask应用。需要扩充状态位时，采用位扩充的方式替代字段扩充。

■ 对于数据量很大的数据结构，比如我们业务当中的Feed数据，保持数据按照代龄进行分层，并且有合理的归档结构，保持活跃数据所在库的规模不过度，对于活跃数据库需要紧急调整的时候，变更代价也是可控制的。



■ 对于数据量非常大的数据需要调整数据结构，那么只能靠数据库架构解决。我们采用的数据库架构都是M-M-S-S架构，需要进行大规模数据迁移，所以会利用备份库进行轮换下线（或半下线）维护变更，通过轮次的数据结构调整，最终达到数据结构完全变更。

**孙立：**如何搭建更加有效的测试环境？测试环境和线上环境毕竟不可能完全一样。

**孙朝晖：**我想分成两个方面来回应。

■ 功能测试环境的建设：功能测试环境主要面临的挑战不是线上、线下系统结构不一致，而是测试数据的不一致，由于数据不一致导致某些边界条件没有测试出来。对于这个问题，需要保持线上、线下关键数据的增量同步，采用小时间粒度的定期日志备份方法解决，同时要保证活跃数据的规模，以便能够控制线下数据库同步的规模。

功能测试环境组建的另外一个要点是关注网络结构的等价性，通过虚拟机系统增加与现网等价的网络拓扑结构。尤其是在交换设备和域名系统上尽可能保持一致，因为有很多问题是由于物理部署结构引起的，从开发的角度很难发现，只有通过一个相对完整的功能测试环境方可发现这一类问题。

■ 性能测试环境的建设：性能测试环境的建设主要用于发现性能问题和容量规划。对于发现性能问题的目标，只要保持数据库服务器尽可能使用物理主机，在物理主机上采用多实例隔离的模式部署，其他服务器可采用虚拟化技术，这样可以保持测试数据的可用性。

需要解决的主要问题是容量规划这项工作，由于线上与线下服务器的硬件类型和规格不一致，网络拓扑结构也不同，也无法完全模拟线上环境。所以我们一般采用的方法是，对服务器进行角色类型划分，同一种角色的服务器在性能测试环境下必须有统一类型的服务器对应，有对应的确定性物理配置，在性能环境测试后，首先安排预上线测试环境，通过七层交换或者应用程序本身的流量复制，把线上的服务压力导流到预上线环境，进行小规模再测试，得出单机的性能数据与性能测试环境的对比，并根据单机的性能测算对比，做出线上的容量规划。按照容量规划

进行系统实施后，根据实际的测试数据再调整，有可能因容量估计过高，服务器能力有冗余，需将服务器标记为空闲，留待后续部署使用。

**孙立：**你如何看待NoSQL的？

**孙朝晖：**我们在建设Feed中心时经过了对NoSQL的充分测试，比较了多种NoSQL技术方案，包括Cassandra、MongoDB、Redis和MySQL HandlerSocket，最终我们采用了MySQL HanlderSocket作为我们的NoSQL解决方案。

NoSQL对SNS类的应用场景来说还是很实用且必要的。因为我们要求的Write throughput很大，数据结构简单，仅需要弱一致性，所以在这种场景下使用NoSQL还是比较合适的。

但对于目前主流NoSQL产品推崇的自动分区等高端功能特性，我们经过测试和权衡还是觉得很难用到实际场景下。一方面是这种自动化分区管理功能使得系统自身的复杂性太高，部署和监控复杂；另一方面是NoSQL产品缺乏专用的备份工具，系统出现单点故障后，恢复的代价太高，给运维造成了很大的压力。

我们最终选择实用MySQL HanlderSocket作为NoSQL解决方案的原因是此方案能够利用MySQL数据库本身的全部分布式特性和管理特性，系统可控性更强，对于运维、数据老化处理、性能调整等方面都有成熟的方向。

对于NoSQL本身的发展和应用方向目前我个人还不是看得很清楚，例如几乎全能的Redis出现，大家会如何使用？在我们的技术体系内使用Redis作为内存缓存，替代了Memcached，没有启用持久化功能。

我个人认为，采用NoSQL技术还是需要经过充分评估的，尤其是与物理架构设计师和运维部门充分沟通，平衡开发和运维的整体工作压力。另外使用之前尽量充分阅读源代码，具备自我技术支持能力后再投入使用。

**孙立：**你如何看待横向扩展（Scale Out）和纵向扩展（Scale Up）？

**孙朝晖：**我个人认为，Scale Out是互联网服务的必由之路，分布式技术是互联网服务不可或缺的技术能力。

但随着服务器规模的不断扩大，运维的压力会越来越大，到了一定系统规模，必须回归

到提升单机QPS能力的方向上来。但是我觉得这里的提升单机QPS不是简单的ScaleUp，而是一个综合优化的过程，比如下面这些实例。

- 我们采用配置了SSD卡的服务器，以本地存储方式替代SAN存储设备作为MySQL数据库的文件存储，提升了MySQL单机访问能力。

- 我们正在测试使用虚拟机的方式，将Java中间件的物理服务器分拆成为多个虚拟机服务器。由于受到JVM的内存限制，我们的单机中间件服务器始终没有得到充分的利用，可以尝试一分多，提高系统并行访问能力。

- 对单机的性能提升，我们更倾向于在单机内部采取应用结构内的Scale Out方式，比如刚才说的虚拟机拆分，还有使用异步I/O替代多线程、对象池技术等，最后可能采取选用内存升级或CPU升级的方案。而Scale Up仅是为了满足某个具体需求采用的综合解决方案的一个组成部分。

**孙立：**大型网站往往会将多种语言进行融合使用，关于这方面，你有什么经验分享？

**孙朝晖：**这是一个互联网行业遇到的通用问题，由于互联网行业采用开源软件比较多，往往由于开源软件的引入，引入了一些非本服务体系内的主流开发语言，比如Scala、Erlang等。这个问题我有两点想说。

第一，在我们的技术体系内，我是比较倾向于减少开发语言使用的。我们每个层次采用的技术都比较确定，Web采用PHP，中间采用Java，基础通信部分采用C语言开发，个人认为引入太多的开发语言和技术，会对技术团队成长和线上业务的可维护性带来很大的问题。

第二，虽然我们的开发技术较少，但仍然有异构平台之间的通信问题。在我们的业务中，还有一部分是用Windows平台C#技术开发的。为了解决异构通信的问题，我们自主研发了基于ProtocolBuffer序列化协议和TCP的RPC通信协议，分别使用C#、Java、C（包括PHP Extension）开发了RPC的通信协议栈，解决了跨平台通信问题。当然选用Thrift也是一个不错的选择，我们没有选用它主要是为了兼容我们IM软件的采用SIP协议。我认为解决异构系统通信问题的需要考虑以下几点。

- 在技术层次之间的通信协议尽量不要选

用某种语言的私有标准，尤其是数据序列化，这样有利于扩展到通用的技术平台。

- 在通信模型上尽量选用通用设计结构。比如我们的进程间通信模型选择了Unix Domain Socket，保持与远程通信方式接口一致，保持一致性，方便于服务的物理架构迁移。

- 在开源软件选择上尽量选择我们技术标准范围内的技术。如果没有找到对应项，一般选择改造现有相似开源软件或者独立开发的方向。由于我们选择Java和C开发技术，开源软件很多，目前来看还没有出现某个方面找不到对应功能的开源软件的情况。

**孙立：**对于大型系统的突发性故障，如何在架构层面帮助快速定位故障的原因？

**孙朝晖：**互联网应用系统由于涉及多服务集成，以致多数的异步调用造成突发性故障很难排查，针对这个问题我们主要采取以下几点策略。

- 每次出现故障时进行分析，准确定位故障的源头逻辑节点与物理节点，统计后作为知识库，标识出每个系统中的脆弱节点，出现故障后，优先挑选脆弱节点排查。

- 在架构进行Review时，尽量保持每个业务流程保证不超过3个物理节点的请求跳转，尽量减少物理层次深度。还有保证在一次调用流程中，级联的异步调用至少2次以下。在所有发出连续同步调用的初始节点上负责统一trace，方便故障排查。在物理架构图中标出这些调用流程中的关键节点，出现问题优先从关键节点开始排查。

- 在物理架构上，尽量明确服务器的角色，在某个角色服务器上部署同类型的服务。对于Java中间件服务器由于服务数目较多，尽量采用虚拟机技术拆分。单独服务器承载独立服务，比较容易定义监控策略并发出预警。

- 每次故障后Review告警与故障的关系，对没有及时告警的部分增加告警监控，告警不准确的部分及时修正，增强监控和告警的准确性。

- 在我们的技术体系内开发了针对分布式应用系统的部署和监控系统，应用服务器会集中的监控服务器汇总系统日志，统一与预警和告警系统集成。📌

责任编辑：董世晓（dongsx@csdn.net）

# 揭秘热文排序算法——SmartHot

文 / 亚男

本文结合我烧网实战，详细介绍了它的SmartHot热文排序算法。

就像Google不会公布其搜索排名算法一样，大多数公司都不会对外公布涉及排名的算法。因为这个算法一旦公布，总会有人去做SPAM。同理，我烧网的热文排序算法也不会对外公布，我们只是向用户提供一些基本的原则。这一算法被我们命名为SmartHot，是在2009年底我烧网改版时设计的，到现在已有一年多时间。这期间我们对算法进行了数次改进和调整，也正因如此，才使得我有机会可以将当时的一些设计思路拿出来与大家分享。希望通过这一算法的产生过程的介绍，给更多的网站、程序设计人员一些启发，同样也希望大家踊跃拍砖，共同学习、共同进步。

我烧网是一个以博客内容聚合为基础的阅读社区，像其他内容聚合网站一样，由于存在海量的内容，因此对内容的提炼尤为重要。在改版之前，我们一直通过人工方式推荐内容，但人工方式有两个缺陷：一个是人力成本过高，另一个是推荐出的内容往往是编辑喜欢而用户不一定喜欢的。

这次改版，我们希望采用众包的方式，让用户为用户推荐内容，从而节约团队的人力成本，同时也能够真正把用户喜欢的内容推荐出来，使用户的互动更加活跃。这个推荐列表就是后来的我烧网热文。从访问统计来看，热文页的PV也远远高于其他页面。而热文的幕后，就是靠这套SmartHot算法支撑的。

在提出要研制一套文章排序算法之前，我们先是对这次改版进行了一个规划，整个规划有一个中心思想，那就是让更多的用户参

与到网站中来，通过用户的参与提高网站内容的质量，然后再把高质量的内容反馈给用户。SmartHot要为这一中心思想服务。后来的实践证明，在这套算法的支撑下，我烧网实现了飞速的发展，最终带来的主要影响有三：一是更多用户参与到社区的活动中来，二是内容质量大幅提升，三是节约了大量的人力成本而且带来了比人工更好的效果。

在这一中心思想的指导下整个团队开始了对热文排序算法的思考。

## 提出问题

首先，以用户行为为推荐的基础并不是说让用户去推荐，因为用户做太多贡献会使网站失去活力。更加明智的做法应该是基于用户的各种单一操作，通过数据挖掘找出最值得推荐的文章。这样就细化成了一个数据挖掘的问题——通过对用户行为数据的挖掘找到好的文章再推荐给全部的用户。

面对这样的问题，团队提出了很多种解决思路。最简单、最常规的当然就是按照文章的浏览量来排序。然而无数经历都证明了这不是一个合理的算法，浏览量的高低只能代表访问者被标题、图片等吸引的程度。如果想要了解用户对文章的真实感受，就得用添柴数（我烧网中用户通过为文章添柴来表达对其的喜爱）来排序。但是，无论是浏览数还是添柴数，都是累计的数字，最终必然会因马太效应产生严重问题，使排在前面的文章永远排在前面，新



文章无法进入前列。这时，将二者结合的思路出现了。

我们把添柴数作为正向的参数，把浏览量作为反向的参数。如果一篇文章被用户浏览了100次、添柴10次，那么就计为0.1分（ $10/100=0.1$ ）。这就意味着，浏览量越大，最终的排序分值越小！这是一个很具突破性的想法，因为我们知道很多同行都把浏览量作为正向参数给文章排序。

问题貌似解决了，但是新的问题也随之产生：大多数用户只看不操作，换句话说很多用户即便觉得文章很不错，也懒得去添柴，这样势必导致很多好的文章因为浏览量太大反而失去排在前面的机会。几番斟酌，我们最终决定放弃浏览量这个依据——**文章得分以添柴数为基础**。

在我烧网中，对文章的其他操作如评论、分享等也类似于添柴，被赋予一定的分数计入文章的得分。

当然，单纯按得分来排序也是不合理的，这与直接用添柴数排序的结果是一样的。为了保证用户喜欢的新文章能够永远排在用户同样喜欢的旧文章之前，我们必须引入时间参数，由用户操作的得分和文章的发布时间共同决定文章的排序值。添柴数（包括其他用户操作）作为正向的参数，文章发布时间作为反向的参数，最终得到的排序用的排序值与文章得分成正比、与发布时间成反比。于是又一重要原则产生了——**文章排序值随时间的前进而衰减**。

分析到这里，团队里的每一个人都欣喜若狂：我们知道算法即将完成了。但是一个尖锐的矛盾又把我们拉回现实：文章得分是很容易作弊的。一个人如果对一篇文章连续添柴，包括换用很多马甲连续添柴，都可以直接获得很多得分。我们可以限制一名用户连续添柴的次数，可是我们无法限制一个用户注册马甲的数量。于是，一个关于用户权重的参数——**用户系数**，出现了。**用户系数与等级相关，等级越高的用户权重越大，刚注册的用户权重极低**。

基本思路产生后，问题也就很明确了，接下来就是根据这些问题确定各个目标，然后解决问题。

## 确定目标

根据前面的分析，可以确定以下目标。

1. 为每种用户操作设定其固有的分值。
2. 设定一个时间衰减的速率。
3. 确立一个用户系数模型，使用户的权重随等级变化而变化。
4. 设计一个用户等级体系。

## 解决问题

### 操作常量（A）

我烧网中用户对文章的操作包括添柴、泼水（添柴的逆操作）、分享和评论等。添柴、泼水是对文章最简单、最直接的表态：喜欢就添柴，不喜欢就泼水。其他的操作所对应的分数应该参考这两项操作。因此将添柴常量定为1，泼水常量定为-1。

**A添柴=1**

**A泼水=-1**

分享也是对文章的肯定。如果一个用户给一篇文章添柴，说明他喜欢这篇文章；如果用户分享这篇文章，说明他不仅喜欢这篇文章，还希望其他人也来看这篇文章。两者比较，分享是比添柴更为强烈的行为。因此分享应当比添柴得到更多的分，这里将其设定为1.2。

**A分享=1.2**

评论则是一个双面性的操作。用户可能为赞同某一篇文章而对其评论，也可能是由于非常反对文章中的观点而进行评论。经过反复思考和研究，我们发现当一个用户对某一篇文章持反对意见而评论时，其实这篇文章是更值得推荐的，因为往往这类文章是那种容易引起争议的或者是那些最容易被炒火的内容。从这个角度上看，无论是正面还是负面的评论，这篇文章都应该被推荐，而且应该比添柴、分享有更高的推荐度。所以我们将评论的操作常量定为1.5。

**A评论=1.5**

### 时间衰减率（T）

时间衰减率的核心问题就是让文章在多长时间

的时间内衰减多大的比例。

一般来说，我们都希望网站上排在前面的内容是新的、热点的。但是这个“新”可以是一个小时以内，也可以是一天以内、一周以内。“新”应该视具体情况而定。访问我烧网的人通常是每天来看看有什么新鲜的内容，因此按天来衰减对于我烧网来说是比较合理的。以 $t_0$ 表示文章的发布时间， $t_1$ 表示当前时间， $T$ 应该是 $(t_1-t_0)$ 的函数，即：

$T=f(t_1-t_0)$ ，其中 $t_0$ 和 $t_1$ 以天为单位。

我们在参考了某网站的Digg版之后发现，即使是排在第一的文章，推荐数也很难超过100；推荐数最多也是100多，极少数的会在200以上，大部分在50~99之间。根据我烧网的文章量和用户量，我们推测文章的得分将很难超过100。

我们列举了一组数据：当 $t_1-t_0=0,1,2,3,4,5,6,7$ ，文章得分（D）=10,20,30,40,50,60,70,80,90,100时，应该设计何种排序方式？

- ①同一天的文章 $t$ 相等，所以 $D$ 越大排名越靠前。
  - ②同样的 $D$ ， $t_1-t_0$ 越小排名越靠前。
  - ③昨天的文章100分，今天的文章10分，20分，30分，40分，这个100分的文章应该还是排在前面。
  - ④7天前的文章100分，今天的文章10分，20分，30分，40分，这个100分的文章应该排在中间。
- ①和②是最基本的原则，很好理解。③和④要解决的是前一个或前 $N$ 个时间周期的超级热门和今天的一般热门是怎样的先后位置。

那么我们大胆设想了两个函数：

$T=1/(t_1-t_0)$  [1]

$T=1/(t_1-t_0)^2$  [2]

将 $(t_1-t_0)=0,1,2,3,4,5,6,7$ 分别代入以上两式，得：

$T=+\infty, 1, 1/2, 1/3, 1/4, 1/5, 1/6$  [3]

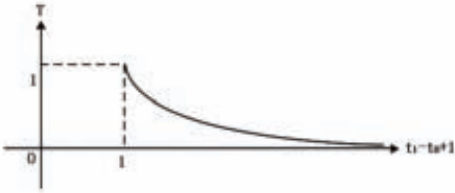
$T=+\infty, 1, 1/4, 1/9, 1/16, 1/25, 1/36$  [4]

以上两列结果显然[3]更合理，[4]让以前的超级热门也没有任何机会，是不合理的。

经过优化，我们需要把正无穷的结果剔除，最终得出：

$T=1/(t_1-t_0+1)$

图1 时间衰减率函数的模型图



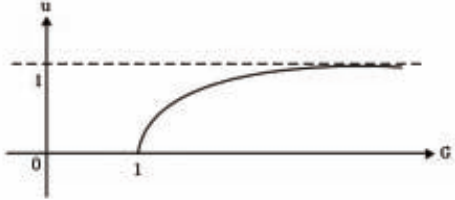
用户系数（u）

关于用户系数 $u$ 的确定，我们首先确定了下面几个原则。

- 新注册用户没有话语权。
- 用户话语权衡量标准是用户等级（G）。
- 新注册用户需要跨过一个很高的门槛才能拥有话语权。但是一旦越过这个门槛，话语权就应该有一个迅速的提升。就是说，当某个用户不再是新注册用户之后，话语权就应该和大家差不多。
- 有话语权的用户之间比较， $u$ 不能相差太大，不然不利于保护新手。总不能让人家新手们添几十次柴还不如老手添一下柴吧！
- 随着等级提升，用户系数 $u$ 的提升应该越来越慢，并且应该有个极限值，不至于让高等级用户太占便宜，这也是为了保证公平性。

根据上面的原则，我们可以看出，用户系数是一个关于等级的函数，这个函数模型应该如图2所示。

图2 用户系数函数的模型图



在图2中，当用户等级 $G=1$ 时， $u=0$ ；当 $G$ 趋进于正无穷时， $u=1$ 。这样的话， $u$ 就永远处于0和1之间，即 $0 \leq u < 1$ 。于是老用户能比新用户占便宜，但也占不到太多。这条曲线还有一个重要点，就是当 $G$ 离开1变为2时， $u$ 应该猛增；而当 $G$ 从大数再增大时（如从10变为11）， $u$ 的增量应该极小。

由上面的模型，我们设计了以下公式：

$u=1-1/(2^G-1)$

用户等级（G）

等级的具体计算公式与SmartHot算法关系不大，这里不展开说明，只谈谈我们对等级的特殊处理。

最初在没有考虑SmartHot算法的时候，我们希望为用户的每一步操作奖励一定的经验，经验达到某个阈值就升级。但是，由于用户等级关系到文章的排序，需要在最初想法的基础上加入一些限制以防止作弊。

对等级的作弊其实很好理解。比如在一般的RPG网游中，玩家可以通过不停地刷怪获得经验以达到升级的目的，放在我烧网中就是可以通过不停地分享文章来达到同样的效果。当用户可以很容易地升级时，作弊者就可以通过注册N个马甲随意地控制热文了。因此在这里我们做出了一定的限制：在用户操作某一篇文章后的一分钟内，再次进行文章操作（即使是对不同的文章）不能获得经验。这样一来，用户每小时、每天能够得到的经验都是有限的，有效地避免了刷经验、刷等级。

算法确定

解决了以上问题，完整的SmartHot算法已经出来了：

$Q=D*T$   
 $T=1/(t1-t0+1)$ （t1、t0单位为天）  
 $D=D1+D2+D3+\cdots+Dn$   
 $Dn=A*u$   
 $u=1-(1/(2^G-1))$   
其中：  
Q——文章排序的最终依据  
D——文章总得分  
G——用户等级  
A——操作常量  
u——用户系数  
T——时间衰减率  
t0——文章发布时间  
t1——当前时间

一点特殊处理：前面曾经提到过，同一用户对同一篇文章的添柴只允许进行一次，但是我们不能按同样办法限制用户对同一文章的评价次数。

因此，为了保证同一用户对同一篇文章操作，该文章只能得到一次加分，我们做了一个特殊处理：当同一用户对同一文章进行了N次操作时，只将其第一次操作产生的Dn计入D，以后的操作皆不影响文章的D值。

总结

至此，我完整地介绍了我烧网SmartHot算法设计的全部思路，可以说这是我们团队整体智慧的结晶，也为后来我烧网的发展起到了重要的作用。SmartHot算法不仅省去了很多人工劳动，更重要的是让我烧网的内容越来越贴近用户。

世上唯一不变的是变化，算法也一样，像Google一样我们也在不停地对算法进行着优化和调整，在即将完成本文时，我注意到版本库中关于SmartHot的修订记录总共有37条。这样算来，从第一个版本到现在，我们平均每个月要对算法改进2~3次，这是连我自己都没有想到的。

现在运行在我烧网上的SmartHot已经跟本文提到的大大不同了，但我们一直坚持着一个重要的原则——只改算法，不改结果。虽然期间很多算法上的重大Bug都影响了我烧网热文的质量，有时也被SPAM钻了空子，但我们这种坚持一直没有变，我们宁可牺牲一整夜的睡眠来改进算法也不愿轻松地修改一下数据库中的结果。因为我们清楚地知道，这样做一次其实并没有什么，可是做过一次之后就将无法停止继续去做。也正是这种坚持，给我们带来了更好的算法、更优质的内容和更多用户的喜爱。

希望以本文与所有正在做和即将做产品设计、算法设计的朋友共勉。



亚男  
我烧网（北京志腾新诺科技有限公司）CTO，  
从事团队管理及核心技术设计工作。

责任编辑：董世晓（dongsx@csdn.net）





在余飞 (Phil De La Zerda) 看来，并行编程对技术人员来说是一个机遇，会像当年Java一样流行。

# 并行编程会像当年的Java一样流行

记者 / 董世晓

在3月25日举行的2011英特尔软件大会上，记者就技术人员所关心的并行编程和多核开发的问题，采访了英特尔软件开发产品部全球销售与业务开发总监余飞（Phil De La Zerda）。

记者：在您看来，开发人员都必须掌握并行编程吗？

余飞：随着对高性能应用要求的增加，简单地依靠提高CPU主频来提升性能已经比较困难，因此多核得到了越来越多的应用。而要充分发挥多核的潜能，就必须使整个软件架构随着多核的发展做出相应的调整，朝着并行的方向发展，因为如果多核上的程序依然还是串行运行的话，其运行效率可想而知——与单核相差不大。

前些年，并行编程主要应用在高性能计算，比如天气预报、能源勘探以及一些金融衍生品等，可以说，还是局限在高端的应用上。这几年形势有了显著的变化，并行编程越来越多地出现在大众主流应用中，比如医疗图像处理、网络游戏、数字媒体等。所以，这些领域的开发人员都非常有必要掌握并行编程。

记者：开发人员对并行编程工具的需求主要是在哪些方面？

余飞：成熟的软件工具，一方面能够为没有经验的开发人员提供一些建议，比如如何在软件程序里添加并行；另一方面可以帮助开发人员做一些预测，比如加入了并行以后，对并行度、性能会提升多少，都会有一个前瞻性的预测，这样可以帮助他们避免走一些弯路。

开发人员在选择并行编程工具时，主要考虑以下几个方面：首先，有一套系统的方法论；其次，应该可以接受比较适当的培训；最后，比较重要的一点就是，可以借助于适合的软件工具去实现想要的功能，因为如果仅仅凭借经验和知识自己动手来做的话，往往花的工

夫会比较大，而且走的路也不一定是对的。软件工具覆盖了开发中的每一个步骤，提供了一套完整的组件去支撑开发工作。无论是编码、测试，还是调优，整个软件工具环境都会提供恰当的帮助。

记者：英特尔有多种不同的并行编程模型，虽然这些模型提供了更多的灵活性，但也增加了程序员的学习成本，带来了选择上的困惑，您怎么看待这个问题？

余飞：的确，我们有几个不同的并行编程模型，比如TBB、LBB等。之所以提供这么多的并行编程模型，也是希望能涵盖并行编程的所有领域。实际情况是，我们针对不同的行业客户，提供了不同的并行编程模型。但确实存在给客户提供的选择过多反而导致他们有时不知所措，所以，我们也正在将这些产品做进一步的整合，以使得用户看到的是更加简单、更加容易理解的方案。此外，我们还会采取一些其他措施，比如培训客户让他们了解不同的产品是针对不同的应用场景的。

记者：对那些有志于投身到并行编程中的开发人员，您有哪些建议？

余飞：对于想投身于并行编程的人来说，最大的建议就是赶快利用一切机会和条件去学习并行编程技术和软件工具。未来，并行编程必将会成为一个趋势，对于个人职业来说也会成为一个核心竞争力。就像我多年前使用Java一样，当时Java很流行，只要懂一点儿马上就会得到一份不错的工作，我觉得并行编程也会像Java一样有类似的情形。随着互联网的发展，人们学习的渠道越来越多，速度也越来越快。网络社区就是一个很好的渠道，在那里你既可以学习知识，也可以分享知识，在互动过程中你就会变得越来越熟练，从而成为这方面的专家。P





主持人：张银奎  
《软件调试》一书作者，从事软件开发和研究10余年，对IA-32架构、操作系统内核、虚拟技术，尤其对软件调试有较深入的研究。翻译（合译）作品包括《数据挖掘原理》、《机器学习》、《人工智能：复杂问题求解的结构和策略》、《观止——微软创建NT和未来的夺命狂奔》等。

# 解救即将被断网的系统

## ——调试补丁安装失败（上）

现代社会给生活提供了很多便利，但同时也增加了很多依赖。比如春节过后的那个月，大家议论的一个热门话题就是电费，几个朋友都说家里的电费比平时高出许多倍，而且想不出原因何在。虽然满腹困惑，但电费还是要交，因为担心被断电。类似的，水、煤气、粮食也都是生活的重要依赖，当然，还有前不久很多人抢购的盐。

除此之外，越来越多的现代人都还有一个新的依赖，那就是网络。对于很多人来说，不能上网几乎与断粮断水差不多。对于从事软件工作的我来说，也是如此，可以食无肉，可以出无车，但是不可以无法访问网络，因为网络断了，就没办法工作了。因此，当收到工作机即将被断网的邮件时真有如临大敌的感觉。为何会收到这样的邮件呢？有些读者可能已经猜到了。

### 紧急邮件

周五一早，打开电脑便看到一封高优先级的邮件，标题旁的红色惊叹号格外扎眼，仿佛在“闪闪发光”，发送者或许觉得还不够，又在标题里加了个紧急（Urgent）字样——URGENT: Patch system(s) to avoid disconnection（快打补丁以防断网）。邮件的正文措辞严厉，要求马上采取行动，修补

系统，否则后果严重，将在4/26/2011切断网络，并可能有纪律处分，断网的日期也特意用红色字体强调，让人看了甚是紧张。

邮件很长，详细描述了要执行的步骤，并且以表格的形式列出了需要修补的两个漏洞（如图1，机器名故意模糊处理），一个是MS10-054，另一个是MS10-070。

通常漏洞和补丁是一一对应的，两个漏洞意味着我的系统中缺少两个补丁没有打，从漏洞编号中的10来看，都是2010年的。2010年的重要补丁到了2011年还没有打好，所以安全系统发火了，发出了断网通知。

但话说回来，打补丁是自动的呀，每次都会强制弹出，倒计时一段时间，就自动安装，甚至自动重启，不管你当时在做什么。那么为什么这两个补丁没有打好呢？查了一下管理补丁的工具软件，的确有这两个补丁，看一下日期，的确是去年的，看一下状态，原来是安装失败了。再尝试一次，还是失败。原来是自动安装补丁没有成功，导致这两个补丁延迟到今天。

### 手工打补丁

周五忙了一天，没有时间解决这个补丁问题。今天是周六休息，考虑到这两个补丁涉及到是否断网的大问题，于是上午便坐到

图1 需要修补的漏洞

Machine Name	Vulnerability
XXXXXXXXXX	<ul style="list-style-type: none"><li>• C - MS10-054: Vulnerabilities in SMB Server Could Allow Remote Code Execution - Windows 7 (x64) (PROD)</li><li>• C - MS10-070: Vulnerability in ASP.NET Could Allow Information Disclosure - Microsoft .NET Framework 4.0 - Windows XP / Vista / 7 (x64) (PROD)</li></ul>



电脑前，着手做这件事。从哪里下手呢？上调试器？有点早，还是先试试普通方法吧。首先想到的是手工安装补丁。

在微软网站搜索MS10-070便可以找到有关的知识库文章KB2416472，然后便可以下载所需要的补丁安装包了，对于我使用的x64系统，补丁文件是NDP40-KB2416472-x64.exe，大小为7MB多。

下载后以管理员身份运行，先是一个自动解压缩的进度条，跑得非常流畅，然后是短暂的序幕（splash），之后显示出图2所示的安装向导对话框。



扫描一下图2中的对话框便会发现有些不对，Next按钮是灰色的，无法继续，而且显然正式安装还没有开始。读一下列表框中的话，“KB2416472不适用”，“或者是被你机器上的另一个条件所拦截了”，这条消息使用了算命先生的语言，万无一失，肯定没错，但是没有什么价值，到底是补丁不适用，还是准备修补时遇到“艰难险阻”了呢？都有可能，让人摸不着头脑。

如果点击Cancel，那么程序便退出了。根据以前的经验，我先找到了刚才解压缩的临时目录，C:\ae069ae1f035efe7517b3834dbc1a7，将其复制到了另一个地方，然后点击Cancel。不然的话，点击Cancel时，这个目录便会被自动删除。

## 事件记录

手动安装没有轻易成功，这也在预料之中，因为自动安装脚本也是反复测试过的，不会轻易失败。接下来要查失败的原因了。首先想到的是查看系统的事件记录，打开事件查看器（Event Viewer），选择Windows Logs，再选择Application，然后浏览刚刚发生的事件。真的有几条都来自MsiInstaller，而且有一条是错误信息，双击打开，如图3所示。



错误信息确认了安装没有成功，针对.NET Framework 4 Extended的KB22417472更新没有能安装，错误代码是1603。接下来提示我们可以启用更详细的日志来定位问题。

搜索Install Error 1603，会在微软知识库中找到编号为834484的文章，描述了导致该错误的很多种可能，所提供的帮助非常有限。值得一提的是，自动翻译系统将这篇文章的标题中的“Error 1603”翻译为“1603年错误”（图4），让人哭笑不得。坦率说，机器翻译的这些技术文章还不如直接读原来的英文。



## 启用安装日志

上面（图3）提到的事件记录提示我们启用安装程序的详细日志，这当然值得试一试。点击那个链接，浏览器打开了一篇名为《如何启用Windows Installer日志记录》的知识库文章，

是中文，但应该是人工翻译的，可读性比那篇“1603年错误”好多了。其中介绍了启用日志的两种方法，一种是通过本地组策略编辑器，另一种是直接修改注册表。我选择了第一种方法，运行gpedit.msc，调出本地组策略编辑器，顺着如下路径层层展开找到Windows Installer的Logging设置（图5）：

Local Computer Policy/Computer Configuration/Administrative Templates/Windows Components/Windows Installer/Logging



双击Logging项，在随后打开的对话框中输入voicewarmupx，启用所有日志信息，其中每个字符代表一个日志选项。

以上操作后，设置便被记录到注册表的如下位置：

键名：HKEY\_LOCAL\_MACHINE\Software\Policies\Microsoft\Windows\Installer  
类型：Reg\_SZ  
名称：Logging  
取值：voicewarmupx

如果直接修改注册表，那么便是修改这个键值。

启用日志后，重新执行补丁安装程序，图2所示的错误信息再次出现，问题仍然存在。于是可以寻找日志文件了。在哪儿寻找呢？图5说在系统卷的临时目录，文件名叫Msi.log。刚才提到的知识库文章这样说：新的日志文件名称是随机的，但以字母“Msi”开头，以.log扩展名结束。若要查找Temp文件夹位置，请在命令提示符处键入下行：

```
cd %temp%
```

看来是在临时目录，对于Win7，只要在开

始菜单的Search program and files编辑框中输入%temp%并回车就可以浏览这个临时目录中的文件了。通常这个目录的路径是C:\Users\<用户名>\AppData\Local\Temp。

但是仔细浏览这个目录中的文件，却找不到符合特征的文件。哪里出问题了呢？联想到启用日志的知识库文章开头有个提示，说该文章不适用于“您正在使用的版本”：

系统提示注意：本篇文章的内容适用于其他版本的Windows（不包括您正在使用的版本）。本篇文章的内容可能不适用您的电脑。浏览Windows 7帮助和支持中心。

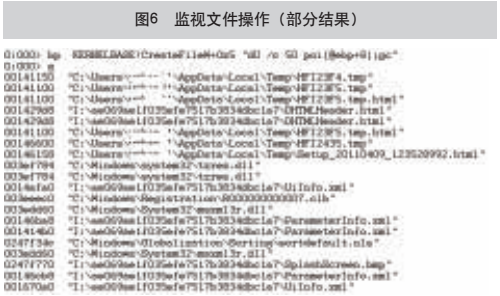
难道日志文件搬家了吗？在系统盘中搜索msi\*.log也没有找到合适的文件，日志文件改名了吗？书中暗表，其实都没有，对于Win7来说，日志文件还在临时目录，文件名还是叫MsiXXX.log。那为何找不到呢？是因为还没有产生。书中再暗表，原来补丁安装程序与普通的安装程序不同，上面的说法对于普通安装程序是对的，但对于补丁程序可能不适用。那该如何是好呢？这下到了上调试器的时候了。

上调试器

以管理员身份运行WinDBG，打开补丁文件目录（上文提到复制出的）中的setup.exe，初始断点命中后，沉吟片刻，思考如何埋设断点。一口茶下肚，计上心头，先监视一下文件操作吧，看看创建和使用哪些文件。于是使用bp命令设置如下条件断点：

```
bp KERNELBASE!CreateFileW+0x5 "dU /c 50 poi(@ebp+8);gc"
```

双引号中的dU命令是用来显示CreateFile函



数的第一个参数，以Unicode方式显示。执行g命令，让程序跑起来，如所预料，条件断点命令持续打印出很多文件名，图6是其中的一部分（路径中的用户名故意隐去）。

根据文件名揣度一下图6中所列的各个文件，会发现有一部分是在临时目录中的.html文件和.tmp文件，有一部分是位于补丁包中的.xml文件。临时目录中的.html文件首先值得一看，实际上，随着程序的运行，图6中列出的Setup\_20110409\_123528992.html会被改名为KB2416472\_20110409\_125430441.html的文件。或者说这两个文件名实际上对应的是一个文件，双击打开观察，前半部分的内容如图7所示。

图7 准备安装补丁的日志文件（部分）



仔细阅读日志文件，看到有下面几行错误信息：

```
OpenFileMapping fails with last error:
6[4/10/2011, 13:10:54]The handle to the
section is Null[4/10/2011, 13:10:54]
OpenFileMapping fails with last error:
6[4/10/2011, 13:10:54]The handle to the
section is Null
```

从这个信息来推测，安装程序曾经调用过OpenFileMapping API，而且调用失败，通过GetLastError()函数查到错误代码是6。

```
0:000> !error 6
Error code: (Win32) 0x6 (6) - The handle
is invalid.
```

表面含义是句柄无效，详情如何呢，不妨断下来看一看。

## 追查文件映射失败

执行.restart命令让目标程序重新运

行，当初始断点命中后，埋下断点监视对OpenFileMapping函数的调用：

```
bp KERNELBASE!OpenFileMappingW+5 "kv
10;dU /c 30 poi(@ebp+10)"
```

双引号中的dU命令是显示第三个参数，因为根据OpenFileMapping函数的原型，第三个参数是字符串类型。

```
HANDLE WINAPI OpenFileMapping(
    _in_ DWORD dwDesiredAccess,
    _in_ BOOL bInheritHandle,
    _in_ LPCTSTR lpName);
```

设置断点后，执行g命令让程序执行，很快断点命中，粗略观察，如果调用来自无关模块，则执行g命令继续运行，这样命中几次后，可疑情况出现了，来自SetupEngine模块IronMan::MmioChainee::GetEventName方法的调用指定的名称参数居然是空字符串，如图8所示。

图8 错误调用OpenFileMapping函数

```
0:000> ln 6911fe88
(6911fe70) SetupEngine!ATL::g_strmgr+0x18
```

单步跟踪OpenFileMappingW函数，果然在调用NtOpenSection内核服务时失败，返回的状态码是c0000024，即STATUS\_OBJECT\_TYPE\_MISMATCH，这个状态码对应的LastError就是日志文件中出现的错误代码6。看来打开文件映射失败是因为指定的名称参数为空串。分别使用ln命令和!address命令搜索这个串的信息（如图9）。

图9 观察空字符串的来源

```
0:000> !address 6911fe88
700 ffd000 is range ffd00000 ffd00000
ProcessParameters: 00401100 is range 00400000 00400000
Environment: 00401100 is range 00400000 00400000
6911fe88 : 00401100 - 00401100
Type: 00000000 MEM_IMAGE
Protect: 00000004 PAGE_READWRITE
State: 00000000 MEM_COMMIT
Usage: RegionUsageImage
FullPath: C:\WINDOWS\WinSxS\x-ww7053344b-c1c7-4266-a012-63994d63d872\SetupEngine.dll
```

从图9中的信息推测，空字符串来自自定义在SetupEngine模块中的一个全局变量。那么这个空字符串是否是导致补丁安装失败的根源呢？为了保证在断网日期前解决这个问题，我会继续分析，但分析的结果只能写在下一期的文章中中了。🔴

责任编辑：董世晓 (dongsx@csdn.net)

## 本期问题：

上期答案是：在Task Scheduler中观察任务的历史记录，即根据文章中的介绍先找到上传任务（WPD\SqmUpload\_XXX或者Microsoft\Windows\Customer Experience Improvement Program\Uploader），再观察History页的信息。

本期的问题是：正文中使用的两个断点命令都没有设置在函数入口，而是设在+5的位置，比如bp KERNELBASE!OpenFileMappingW+5和bp KERNELBASE!CreateFileW+0x5，请解释这样做的原因。

## 编者说明：

- 投稿信箱：  
contest@csdn.net
- 联系方式写在一个单独的TXT文件里，包括以下几项：
  - 1) 姓名
  - 2) 工作单位或学校
  - 3) 电话联系方式
  - 4) 邮寄地址
  - 5) E-mail地址
- 解答提交时间，最好早于当月15日。





### Evoluent VerticalMouse

如今有很多人体工程学鼠标和其他的电脑设备，都有助于缓解手腕伤痛，防止肌腱疾病。VerticalMouse这款鼠标的不同之处在于握住鼠标的手旋转了90度，从而整个小臂和手腕都处于更加自然和舒适的位置。如果现在身边有个普通鼠标，不妨感受一下竖握鼠标，手腕确实会轻松很多。

## GEEK 产品



### evoMouse

Cellulon所提出的evoMouse，能感应到手指的动作，只需要在evoMouse的感应范围内划拉手指，就能完成相应的指令。鼠标动作包括光标控制、点选、双按、右键单击、拖曳、卷动、旋转、放大缩小、前一页与下一页，甚至还可以当绘图板。



### X-mini Capsule 扬声器

手机、MP3、平板电脑、笔记本电脑等移动设备，也许某一天需要响亮得让所有人都听到。X-mini Capsule扬声器只有鸡蛋大小，但放出来的声音，却足以让整个大会议室里的众人听得清清楚楚。仅需接上音频线，采用USB充电，不充电的情况下，就可持续工作4小时以上。

### Sauna Boy

北方的春天风大干燥，办公室的桌上摆一个Sauna Boy小型空气加湿器，不但可以解决干燥问题，还可以作为小小的装饰品。Sauna Boy相当小巧，高度不过8公分，只要加水之后，通过USB与电脑相连即可使用。



### Q-SIM

这款独一无二的设备可以让iPhone 4同时装上两个SIM卡，使用两个不同的号码，自动在两张SIM卡间切换。安装的时候，将一头插入iPhone电话原本装SIM卡的位置，留在机身外的另外一头有两个卡槽，可以安装两个SIM卡，并利用保护壳隐蔽外面的卡线；切换功能可以使用Q-SIM的应用进行设置。虽然样式不是很完美，但功能上几乎为iPhone 4增加了一个新特性。



### Verizon 4G LTE Network Wifi Hotspot

Verizon运行着全美最大的4G网络，但并不是所有的移动设备都能够接入。为了能随时实现不同设备的接入，这几款小卡片样式的设备可以建立起一个WiFi热点，让不同的WiFi设备接入到4G网络中。



# 新书上架

## Python学习手册：第4版

作者：Mark Lutz  
译者：李军 刘红伟  
出版社：机械工业出版社

Python初学者阅读此书，可以广泛了解Python语言的核心内容。书中每章都附带“练习题”，提供实例的训练和测试。



相比前三版，本书增加了5章新内容，第27章提供了Python类和OOP的渐进教程，讲解了这方面的基础知识，并且说明如何在较大、实用的程序中，综合运用类的概念。后面新增的4章，是并非每位Python程序员都需要深入了解的Unicode文本和元类的细节。第36章提供了Unicode和字节字符串的详细介绍，概括了Python 2.6和Python 3.0中字符串和文件的区别。第37章介绍了特性这样的管理属性工具，并给出新的描述符介绍。第38章用示例全面讲解了函数和类装饰器。第39章介绍了元类，对比了它们和描述符的区别。

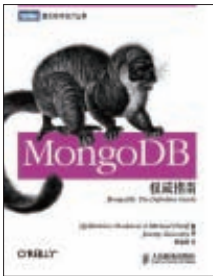
## MongoDB权威指南

作者：Kristina Chodorow  
Michael Dirolf  
译者：程显峰  
出版社：人民邮电出版社

随着Web应用的数据膨胀和复杂度增加，MongoDB受到大家的广泛关注。它并不迎合所有场景的需求，而是在功能和复杂性之间取得很好的平衡，将原先十分繁复的任务大大简化。MongoDB具备支撑今天主流Web应用的关键功能：索引、复制、分片、丰富的查询语法，特别灵活的数据模型，同时并不牺牲速度。

本书涵盖了上述MongoDB的关键功能，讲解了从安装、共享到备份式集群的产品化部署一整个应用流程，并提供了流行客户端API和高级管理主题的实例分享。

两位作者均来自开发并支持开源数据库MongoDB的公司10gen。通过他们的权威解读，你会了解面向文档数据库的诸多优点，并发现MongoDB如此稳定、性能优越甚至能够无限水平扩展背后的原因。



## HTML5与CSS3权威指南

作者：陆凌牛  
出版社：机械工业出版社

本书作者详尽讲解了HTML5和CSS3的所有新功能和特性；所有知识点都紧跟HTML5与CSS3的最新发展动态；每个知识点都配有小案例，而且还有两个综合性的案例。



全书分为三大部分，第一部分介绍了HTML5，包括各主流浏览器对HTML5的支持情况、HTML5的结构元素、表单与文件、图形绘制、多媒体播放、本地存储、离线应用、通信API、Web Workers、地理位置信息获取等内容；第二部分阐述了CSS3，涵盖选择器、文字与字体的相关样式、颜色的相关样式、盒的相关样式、背景与边框的相关样式、布局的相关样式、UI的相关样式、Media Queries的相关样式、变形处理、多媒体和动画等内容。第三部分则以迭代的方式讲解了两个实战案例。

## 云计算解码：技术架构和产业运营

编著：雷葆华 饶少阳 等  
出版社：电子工业出版社

本书梳理了云计算的基本架构，阐述了云计算对社会的影响，并探讨云计算未来的发展趋势，提出了“云计算真正的技术创新，在于IT服务化和规模化服务运营”的观点，从电信运营商的角度对云计算的关键技术提出了见解。

目前国内正处于云计算蓄势待发的阶段，人们对云计算的真实已经接受，在云计算的技术、产品等方面也有了深入的理解，但对云计算的总体架构、业务和技术的关系，云计算运营中的若干关键技术和影响因素则了解较少，也鲜有这些方面令人满意的资料可供参考。这本书适逢其时，从操作系统的视角全面解读云计算，阐明了云计算不是简单的技术应用，而是新的IT软件架构技术。阅读此书，既能从宏观视角全面认识云计算，也能从微观业务实现的角度接触云计算，更多了解云计算应用与运营方面的内涵。





C#与.NET 4高级程序设计 (第5版)

作者: Andrew Troelsen  
译者: 朱晔 等  
出版社: 人民邮电出版社

这是被誉为“C#圣经”的经典巨著，任何关于.NET的细节都可在其中找到。因语言生动流畅、剖析深入、涵盖全面，本书广受推崇，畅销不衰，曾经获得Referenceware编程图书大奖，并入选Jolt大奖提名。书中探讨了C#语言和.NET平台的各种特性，包括重载运算符、指针、泛型等高级功能和CIL、远程处理、Windows Forms、ASP.NET、ADO.NET等技术，作者用IL来剖析语言特性，犀利透彻，语法糖暴露无遗。新版更是透彻阐述了C# 2010和.NET 4新功能，包括动态语言运行时（DLR）、任务并行库（TPL，包括PLINQ）、ADO.NET实体框架（包括LINQ to EF）、扩展的WPF API，以及改进的COM互操作。



数字图像处理原理与实践: 基于Visual C++开发

编著: 左飞 万晋森 刘航  
出版社: 电子工业出版社

本书讲述了在Visual C++环境下数字图像处理编程的技术和方法。全书共分14章，对数字图像处理领域中的核心话题，如色彩空间、图像文件格式、图像的点运算、图像的几何变换、图像的增强处理、边缘检测和轮廓跟踪、图像形态学、图像分割、图像加密等内容，都做了详细的介绍。



比较同类书籍，本书有如下优点：与集成开发环境深度结合；深入剖析MFC机制，并结合丰富的图像处理与编程实现内容；书中穿插大量实例，大都为时下流行的图像处理软件的某个具体模块或部分的实现，实例原型十分常见；作者编写了一个扩展性良好的数字图像处理算法试验平台Magic House。不仅完整实现了普通数字图像处理软件应具备的基本功能，并且框架清晰、容易理解。

云计算核心技术剖析

作者: 吴朱华  
出版社: 人民邮电出版社

随着云计算的发展，获得信息和应用将如同用电一样便捷，成本低廉。企业也可以通过将整个IT基础设施外包来降低运营成本，专注主营业务。

本书将带领大家走进云计算全景世界。整本书分为四部分：一为理论篇，主要介绍云计算理论方面的知识；二为产品与技术篇，深入剖析多个顶尖云计算产品的实现，介绍了云计算中非常重要的系统虚拟化技术和安全方面的机制；三为实践篇，将选择云的核心模块之一——分布式数据库作为实践的方向，并以YunTable 这个云时代的BigTable 为例，给大家演示如何手动编写和设计一个分布式数据库；最后是展望篇，邀请读者一同来猜想和预测云计算和整个科技事业未来的发展。



全球排行榜

Amazon	
1	In The Plex: How Google Thinks, Works, and Shapes Our Lives
2	iWoz: How I Invented the Personal Computer and Had Fun Along the Way
3	Kingpin: How One Hacker Took Over the Billion-Dollar Cybercrime Underground
4	Head First Java, 2nd Edition
5	Beginning iPhone 4 Development: Exploring the iOS SDK
6	Programming in Objective-C 2.0 (2nd Edition)
7	The Web Designer's Idea Book, Vol. 2: More of the Best Themes, Trends and Styles in Website Design
8	Introducing HTML5 (Voices That Matter)
9	Cracking the Coding Interview
10	Programming in Objective-C 2.0
天珑书局（中国台湾）	
1	前进Android Market! Google Android SDK实战演练
2	Google Android 2.X应用程序开发实战
3	ASP.NET 4.0专题实务II: 范例应用与4.0新功能
4	Android案例开发完全讲义
5	深入浅出Android系统原理及开发要点
6	iPhone创意程序设计家（适用SDK 3、SDK 4）
7	Google! Android 2手机应用程序设计入门
8	商业智慧
9	深入浅出Java程序设计
10	鸟哥的Linux私房菜：基础学习篇
第二书店	
1	软件框架设计的艺术
2	项目百态：深入理解软件项目行为模式
3	设计原本：计算机科学巨匠Frederick P. Brooks的思考
4	Python灰帽子：黑客与逆向工程师的Python编程之道
5	深入理解计算机系统（原书第2版）
6	项目管理修炼之道
7	算法导论（原书第2版）
8	Python基础教程：第2版
9	程序员面试宝典（第三版）
10	Android应用开发揭秘

# 一生的学习

文 / 李明

有三件事影响了我18岁以后的人生。

在写下这篇文章题目时，我正好在读一本同名的书——印度著名哲学家克里希那穆提的《一生的学习》。封底有一句话让我深受震动：“我们配做父母吗？想想我们曾经受过的教育，想想我们是如何一天天变得平庸的。”当一个人活到三十多岁将为人父母时，看到这样的责问总不免陷入真切的愧疚和对过往的回忆中。

从大学开始，“平庸”这个词就毫无顾忌地刻入我的人生印章，给每天的工作生活盖上一个触目惊心的红戳。拒绝平庸的斗争日日上演，虽至今未有胜绩但仍不懈于心。我想说的这三件事也与此有关，与其说题目借用了书名，倒不如说是这本书启发了我写这篇文章。如果有感于“自我的平庸”，渴望通过“一生的学习”来找出一条“心灵救赎之道”，可以读一读我的故事。

## 编程序

1996年，我从杭州高级中学保送进入复旦大学计算机系学习。而此前我的理想是做一名物理学家，像爱因斯坦那样。我从高中起就开始自学《相对论》方面的书籍，并立志要在26岁前成名。记得当时我阅读了很多有关爱因斯坦的传记，来到北京后，还专门去拜访了《爱因斯坦文集》的中文译者许良英老先生。这些事情在很长一段时间里激励着我，照亮我过着有梦想的生活，即便后来没能实现，也是充实而有益的。

我进入大学时正值互联网从美国传到中国，那时候起的名字叫“信息高速公路”。学计算机在那个年代很“时髦”，IT男也不像现在这么多，还是个貌似前卫的称呼。当时我所在的杭高已经开始有计算机课程，教我们用Basic语言打印杨辉三角形和九九乘法表。大一开始学习《C

语言》，我的“程序人生”就这样开启了。

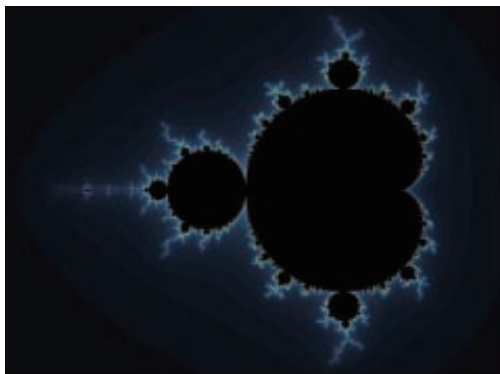
直到今天，编程仍然占据我工作生活的重要位置，也带给我极大的乐趣。后来发生的另外两件事，虽然也花费了十年时间，但从趣味的角度，都不如编程。这十年来，我也接触了很多计算机系的学生，他们中的大多数人都学了类似的专业课程，却只有很少的编程经验，更不用说编程乐趣了。大学毕业时，虽然我的编程水平也不算特别优秀，可是我已经有过几次快乐的体验。

我的第一个快乐体验来自用C语言实现了分形算法，在显示器上成功地画出了“曼德尔布罗



意气风发的亚嵌教育研究中心教学总监李明

曼德尔布罗特集合



特（B.B.Mandelbrot）”分形。那是一个非常美妙的图形，也是我第一次把计算机工具和数学理论相结合的尝试。在没有计算机之前，人工画出这个图几乎是不可能的，因此从这个编程体验开始，我终于又在一个新领域找到了新的梦想。

后来的大学四年间，我又经历了几次这样的成功喜悦。我特别希望那些读计算机专业却不能发现其中乐趣的学生，可以尝试自己做一次。例如：用回溯法实现“八皇后”和“迷宫”问题，用inode实现一个类似Ext2的文件系统，用Lex和Yacc实现一个简单的C compiler，用C++和STL库实现一个XML parser等。

编程序除了带给我快乐以外，还有一些别的收获。例如培养了逻辑思维能力，学会了设定目标和分解任务，知道了按照模块层次结构来进行函数接口设计。这些表面上看起来是编程技术领域的事情，而实际它们就是在程序员世界中的一个虚拟公司管理，例如模块功能划分就好比部门组织结构，函数接口设计就好比岗位职务说明，接口内部实现就好比业务工作流程。

世界万事万物道理是相通的，在一个领域学习获取的专业知识，最终都可以抽象提取出一般规律，构建成常识模型，从而在另一个领域的学习中受益。如果编程能学得好，以后做管理也有前途，这是我从“编程序”到开始“办公司”之后，收获的一个意外发现。

## 办公司

2002年我已从上海到了北京，在清华读研究生二年级。那一年，我开始尝试做一些编程以

外的事情。12月的一天，我像以往一样，在晚饭前先上水木清华BBS看帖子，很偶然地看到一个技术小组正在招募爱好者加盟的消息。与众不同的是它提出的技术理念吸引了我——“自由、协作、创造，一切为了明天”。这个小组就是AKA，这个帖子就像当年Linus Torvalds在邮件列表里面发布的那个消息在后来开启了Linux开源项目一样，两年后同样成就了一家公司。

AKA取名自童话《尼尔斯骑鹅历险记》中那只领头大雁的名字。就是这样一个带有童话色彩的名字，吸引了大批来自清华、北大、中科院等院校的技术爱好者，带着理想主义的情结，纷纷参与进来。可以说中国最早的Linux爱好者和开发人员，多少都得益于这个小组的无私贡献。例如《Linux设备驱动程序》译者魏永明、利索脚lisoleg网站创办者章峰、江苏梦兰龙芯产业化基地总经理张福新、清华大学计算机系陶品老师、陈渝老师都是当时AKA小组的杰出代表。

我参加的嵌入式小组是AKA小组的一个分支，专注于嵌入式技术的推广和传播。当时我的硕士课题恰好也是与嵌入式Linux方向密切相关，我原本的想法是希望多认识一些技术牛人，这样学习起来快一些。可就是这样一个简单的想法，却影响了我后来近十年的时间。

初期小组的工作只是组织技术讲座和沙龙，写一些普及型的技术文章。后来一些公司注意到我们，主动找我们希望一起做技术推广，有的还捐赠了一些嵌入式开发板，这里面就有Intel、ARM、龙芯这样的芯片公司。设备多了就需要地方保管和存放，我们找了一间平房，美名之“AKA嵌入式开放实验室”，全部免费对外开放使用。房子有了后，自然就要交租子买桌子，我们小组成员中有钱的凑了两三万元钱，AKA Embedded OpenLab就这样挂牌成立了。

2003年底，小组的理想主义产物“开放实验室”走到了尽头。当初凑的钱花完了，如果再没有生财之道，就只能退房走人。正是在这个时候，魏永明帮助了我们。他邀请我们，借用我们的场地，做了几次商业培训，给我们分了一些钱，刚好解决了那时的房租问题。2004年，在那样一个基础上，我们成立了一家、也可以说当时国内第一家以嵌入式Linux为主要内





容的培训公司，这就是现在的“亚嵌教育”。

当时小组常委会的16名成员都列入了股东名册。公司注册时，工商局的人非常惊讶，估计他们很难想象16个股东在一起是如何运营公司的。从“程序员”到“办公司”，除了怀抱着同样的热情和执著外，如何搞好经营管理，对我们来说是完全陌生的，不过好在人多，万事有商量。

对于我们这样一批“品学兼优的知识分子创业者”来说，公司这个概念并不容易理解。我们一直在不断学习、加深认识，就好像来亚嵌的学员，有的写了很多年C程序，却不能区分函数栈和栈指针一样。这些年来，我从公司董事长，到总经理，再到教学总监，体验了各个角色的职责和能力要求，也经历了任何成长中的公司必须经历的波澜起伏和人事离合。

从研发工程师转型到公司管理者，是一件颇不容易的事情，这需要打破许多固有的惯性思维。例如工程师会考虑哪儿有Bug并集中精力来解决，管理者却需要首先去寻找机会而不是关注问题；工程师会热衷于弥补技术短板以使得自己的技艺更为全面，管理者却需要首先去发现每个人的长处并知人善用；工程师会满足于自我的成长和技术的精进，管理者却需要关注别人的成长从而使团队更加完美；工程师会经常思考用何种流程和方法来消灭错误，管理者却需要能够容忍错误甚至是鼓励犯错；工程师会希望依靠几个晚上的通宵作战就取得战役的胜利，管理者却需要

知道事业成功是坚持几年甚至几十年的结果……

虽然当初我加入AKA时，并没有预想到后来会走上创办公司这样一条道路，但在这路上的八年来，却是收获了一些别样的感受。这些感受总结起来，就是阿里巴巴马云说的，创业者需要学会和“人”打交道。

## 做教育

2006年我从清华实验室出来，开始投入到亚嵌教育全职工作。正是在那一年，亚嵌开始从短期技能培训转向针对大学生就业的长期职业教育。有一次因为参加一个论坛，需要填写个人行业背景，我在考虑自己是选择IT行业，还是教育行业时，才意识到自己不再是以前的IT男，而是一个教育工作者了。

嵌入式技术在这五年间以从未有过的速度发展起来，最好的证明就是曾经濒于倒闭的Apple公司一跃成为全球市值最大的IT企业，无论是iPod、iPhone、iPad，还是App Store，都可以说是嵌入式软件和硬件的完美结合。随着新东方、安博、学而思等一批教育培训机构的成功上市，教育行业越来越得到风险资本的青睐，我们的亚嵌教育也开始有了上市的梦想。

现在来亚嵌应聘的老师中很多都有开发经验，但没有教学经验。我常问的一个问题是“你知道搞开发和搞教学最大的不同是什么吗？”有

的说是老师更需要口头表达能力，有的则说是更需要沟通能力。在做了五年教育工作之后，我的切身感受是：搞开发是实现自己的梦想，而做教育是帮助别人实现梦想。

唯有把价值观从关注自身转移到自身之外，教育工作者才能获得成功。方法和技巧都可以在教学过程中学会，但爱与善良却是从事教育的人必须先天就具备的品质，无法企求通过后天的训练和辅导获得。如果有人发现周围人评价自己似乎先天就具备了这两项品质，那么最合适他的职业应该就是从事教育。

正确的教育所关心的乃是个人的自由，唯有个人的自由，才能带来个人和团队的真正协作。教育的任务，在于帮助学生了解自己，尊重他人，成为完整的人，实现并享受自由。自由源于对自我的认识，正确的教育最终应该能够帮助学生发现他最感兴趣的事物。所以说，真正的教师不只是一个传授知识的人，更应该是一个唤醒智慧的人。真正的教师也必然能够认识和教育自己。对于他们而言，教育本身就是一种生活方式，除非一个人具有从事教育的热望，否则他就不应该做一名教师。

当初AKA小组的技术理念“自由、协作、创造”中，正是“自由”这个词深深吸引了我。如今十年后我再一次通过教育工作，又重新认识了“自由”的内涵，我把它称之为一条“自由学习之路”。

真正的学习之路应该是自由的，也只有自由学习才是最理想的学习方法。学习的自由是建立在这样几个前提之上。

首先是目标，要清楚自己想要达到的学习目标。学习中的不自由，大多是因为目标不清导致的，因此正确的学习应该要从弄清楚目标开始。设定目标的能力是实现自由过程中最需要也最容易被忽视的一种能力。

其次，在这种“自由”选择之后，就必须承担起责任，包括主动发展自我和积极沟通协作的责任。在亚嵌教学过程中，我发现很多不够优秀的学员，都是因为不清楚这两项是必须的责任。他们简单地认为那是又一次的选择，即我可以选择不去主动，或者我可以选择不去协作，由此而造成学习效率低下是非常可惜

的。只有在学习过程中，学习的主体充分意识到自己对于其他人（同学和老师）所承担的责任时，教学活动才会真正有效。

自我管理的责任，可以这样理解：对一个人发展负有最大责任的并非别人，而是学习者自己。负责管理你的也不是老师或者老板，而是自我。沟通协作的责任，则意味着一旦你是某个领域的专业人士，你就具有去教导别人的责任。这既是社会得以延续发展的源动力，也是个人影响社会、回报他人的最好方式。

“自由、协作、创造”是亚嵌的教育理念，同时也昭示着一条正确的学习道路。只有通过这条道路，我们才能要求学员今后去承担更大的责任，而不是离开亚嵌就无所依靠，也只有通过这条道路，亚嵌的学员才能完成未来社会对他们的更高要求，那就是“创造”。

自由的选择、协作的责任、创造的终极目标，从更高的层次上可以说就是每个学习者的人生。

## 结语

这三个故事都有一个主题——“学习”。学习是一生的话题，我们也需要“一生的学习”。在我看来，学习是能够让自己更接近真理、远离平庸的唯一道路，也是每一个平凡的人都可以身体力行的。

嵌入式技术领域在过去的十年变化之巨，超过了以往任何时候。从2000-2010年，嵌入式的学习书籍，从屈指可数的几本发展到多达数千种。嵌入式的开发板，从一块价格5000多元降低到了不到500元。下一个十年的技术将会以更迅猛的速度发展，如何应对未来变化，要求我们这些从事技术领域开发工作的人们，必须具备更强的学习能力。

“求知若饥，虚怀若愚”（Stay Hungry, Stay Foolish），这是乔布斯在斯坦福大学演讲时提到的一句话。我很喜欢这句话，把它作为此文的结尾，也谨作为对于那些和我一样以学习为乐的朋友们，在学到中年时的一点纪念。📌

Mac OS X 背后的故事（一）

# 力挽狂澜的Ellen Hancock

文 / 王越

从本期开始，我们将在杂志上连载一系列关于Mac OS X发展历史的文章。本系列将为大家介绍Mac OS X的发行版本、技术历史、相关人物等内容。本文是系列连载的第一篇。



Ellen Hancock曾任苹果公司技术总监

故事还得从20世纪90年代说起。Ellen Hancock是本文的主人公，也是一位女英雄。她因在IBM的经历而被人们所熟悉。1966-1995年间，Ellen Hancock在IBM共工作了29年。1985年，她成为IBM的副主席。在1986-1988年间，Ellen Hancock担任过IBM通信产品的主席，并在1992年被选为资深副总裁。1995年9月，她被时任

美国国家半导体（National Semiconductor）CEO的Gil Amelio忽悠，跳槽来到这个企业，做执行副总裁。她在这里带领团队完成了CompactRISC架构，这个架构事后成为ARM7系列的前身。很多人早已经把她忘了，也很少有人能够在回忆时将她和Mac OS X联系起来。但事实上，她是让苹果放弃Copland转而购买NeXT的关键人物。

早在1994年，Gil Amelio就找好了下家Apple，成为Apple董事会的成员。1997年2月，Gil Amelio从National Semiconductor辞职，并成为Apple的CEO。为了紧跟老板的召唤，Ellen Hancock再次被忽悠，来到了当时危机四伏的Apple。这时是1996年5月，为什么是危机四伏呢？还得从早先的事情说起。

20世纪80年代，卖可乐的John Sculley成为Apple的CEO，随之Steve Jobs被轰出Apple。毕竟可乐和计算机不是一回事，因此不管是硬件还是Mac OS，整个公司的开发项目越来越受阻。而且由于先天的不足，Mac OS从诞生之初就不具有一个现代操作系统所应有的特性。所以，在1987年，开发下一代操作系统的计划呼之欲出。具体的规划是，把新的系统所需要的功能，写在一堆卡片上。短期可实现的目标，比如增加颜色支持（当时计算机仍是黑白的），写在蓝色的卡片上；长期的目标，比如多任务功能，写在粉色的卡片上；而在可预见的未来都无法实现的长期



的目标，比如加一个纯物件导向的文件系统，就写在红色的卡片上。在这样的思路下，Mac OS的开发团队马上就被分成两个组，一个叫蓝组，目标是在1991年，发布一个关于Mac OS的更新版本；另一个叫粉组，和蓝组同时工作，计划在1993年，发布一个全新的操作系统。

1991年5月13日，蓝组顺利按时完成开发任务，发布了Mac OS 7（一般被称为System 7），而粉组却没做出什么有实际用途的东西来，因此接连跳票。而且，由于Mac OS 7的发布往往缺乏人手，为了保持正常发布，常常需要从粉组抽调人员参加蓝组的开发，再加上Apple当时重心放在了和IBM等公司的合作上（Taligent项目）而不是在粉组上，最终导致了粉组项目夭折。而本来Apple指望和IBM合作的Taligent项目能开发出一个可用的新系统，但后来IBM不跟Apple继续玩了，因而Taligent的果子又吃不到，Apple相当郁闷。这时由于Mac OS有先天不足（单任务，没有内存保护），再加上Apple以及第三方软件的无限量增加（在这段时期，单Apple自己就已经加入了QuickDraw、PowerTalk、QuickTime等软件和技术，每一个都比Mac OS本身来得大），Mac OS的问题终于大爆发。上个世纪90年代，Mac OS给人的印象就是很不稳定、经常崩溃，同Windows 95留给PC用户的印象差不多，甚至更甚。

Taligent项目挂掉后，Apple自己尝试过十多个不同的内部项目，但大多没做多久就夭折了。而这时正是Windows NT走向成熟的关键时期。眼看着日子逐渐变得不好过了，Apple开始重新开始考虑建立下一代操作系统的事情。1994年，Mac OS 7.5 (Mozart) 发布后，Apple推出新规划，建立一个全新的操作系统，以Copland命名（纪念 Aaron Copland，Mac OS的发布以音乐家名字命名，和Mac OS X后使用猫科动物名字很不一样），这个项目将有一个全新的内核，具有类似Windows NT 内核的所有高级特性，而老的软件都当作独立的进程模拟运行。这个项目时间紧、任务重，1995年3月公

布计划，预期1996年发布。而Copland后的版本Gershwin（纪念George Gershwin），预计1997年发布，将重写Mac的所有系统主要部件，以适应新内核的各种特性。

Copland将使用微内核技术，只做任务和内存分配。除此之外的所有功能，比如文件系统、硬件驱动等作为微内核上的服务运行。而Mac OS的所有用户界面功能将成为一个独立的框架，称为蓝盒（Blue Box，今后介绍Mac OS X时，我们还会遇到这个词）。所有的任务相互独立，占用独立内存，也可以用IPC相互交流。学过操作系统的人都知道，微内核是当时的一个热词，一个系统只有被称为微内核才可被看作是先进的，当时还有针对Linux系统的著名的Tanenbaum-Torvalds笔战。但事实证明，所有本来想做成微内核系统的成功项目都放弃了原先的设计（包括NeXTSTEP、Windows NT），因为这种类似Mach微内核的系统往往难产，GNU/Mach + Hurd之类的项目做到现在经过了20年，仍未成事，一年内搞一个微内核系统谈何容易。

微内核还没搞成，Apple几乎所有开发组的成员都来添乱。大家都说自己做的东西很重要，一定要加入Copland开发组，所以QuickDraw GX、OpenDoc之类的开发组产品成为新系统的核心组件，甚至类似用户界面皮肤之类的开发组都来凑热闹，马上就使Copland成为一个无法维护的项目。开出的计划越来越长，项目越来越多，但相关进展越来越少，完成速度越来越慢。即便做出了产品，连测试人手都不够。1995年就有工程师指出，在1996年发布Copland纯粹是幻想，能1997年发布就不错了。

1996年，Gil Amelio已经掌权。在苹果电脑全球研发者大会上他开心地宣布，传说中的Copland，也就是System 8的开发版会在当年夏天发布，而正式版在秋天就可以送到每位用户手上。时任TidBITS编辑的Matt Neuburg有幸见到了这个传说中的系统。令他大吃一惊的是，这个系统在当时只能打开或关闭文件，而无法

对文本文件进行编辑，甚至所有用户界面的文本框都不能输字。哪怕什么都没做，整个系统也会随机崩溃，而崩溃甚至会造成文件系统损坏。参加演示的苹果员工，则需要不断守在旁边，他们的工作是不断地格式化已崩溃的计算机磁盘，然后重装系统。那年夏天，第零个测试版送到一小簇不明真相的开发者手中，把那些脆弱的没见过世面的人吓得半死。就连Apple内部都开玩笑说Copland的正式发布日期可能得推迟到2030年。

Gil Amelio心急如焚，希望Copland快点走到正道上。作为Gil Amelio永远的好朋友，Ellen Hancock就在这个乱糟糟的时候来到了Apple。她的职务，正是担任这个乱糟糟项目的负责人。她亲自下访各小组体察民情，了解情况。毕竟在IBM干了近三十年，她靠自己过人的判断力在2~3个月内便得出结论，Copland这个项目是没有指望的，就按目前Apple这样的状态，Copland永远都不可能发布，还不如早点取消了好。在短期内，先把Copland中的一些有用的成果一点点合并到老的Mac OS中，并且抓紧从外部购买一个全新系统来满足Apple的需要。正是她的这个结论，结束了Apple长达五年的纠结，使公司重新走向正轨。整个PC的黄金时代已经过去，Apple想要翻身，还有很长一段路要走。Gil Amelio支持了Ellen Hancock的计划。1996年8月，Apple取消Copland项目。开发预览版的CD封套都已制完，每个邮包上的地址都已打印就续，而CD却从未曾制出。

1996-1998年是Apple最混乱的几年。在商业上，有一阵曾传出Apple要被Sun收购的消息。更有意思的是，《连线》杂志在1997年的六月还发表了一篇文章，名为《101种拯救Apple的方法》，其中一条说最好的方式是Apple让自己被Motorola买下，成为Motorola的一个部门，做PowerPC系列产品。以当时的眼光来看这些建议非常讽刺好笑，以今天的眼光看更为好笑。而Ellen Hancock在这段时间内的出色工作，成功地挽救了Apple。

首先，Ellen Hancock的对内政策是继续Mac OS 7.5的开发工作，一步步把Copland中的技术并到7.5中。同时，也大量购买第三方的系

统增强包，包括插件管理工具、层次化菜单等技术。Apple把它们买过来，整合到现有的系统中。整个老系统在新系统尚未完成的时候不断更新，至2000年已出到9.0版，尽可能地留住了老用户。并且，前面提到的蓝盒（Blue Box）也作为后来新Mac OS X系统的一部分，支持用户运行经典Mac OS的程序。

而对外政策更是一个大手笔。Ellen Hancock协助Gil Amelio在Apple之外找寻操作系统技术。在IBM和Microsoft合作Big Blue的经验告诉她，购买一个操作系统的使用权问题多多，最好的计划是整个一并买下来。因此，Gil Amelio开始和当时看好的Be谈，却因价格问题没有成功，最终转而收购了NeXT。而Apple合并NeXT后，NeXTSTEP就演化为Rhapsody，并最终成为Mac OS X。这些事情我们今后会详细再谈。

买完NeXT后，Steve Jobs执政，Gil Amelio因任CEO期间Apple亏损严重而被炒。Steve Jobs把信得过的人（很多是前NeXT员工）拉拢到周围，开始新政，而同Gil Amelio有关的Ellen Hancock则在人事变动中被疏远。Steve Jobs甚至在很多场合称她为“笨蛋”。Ellen Hancock最终于1998年主动辞职。事后同Gil Amelio以及Apple的创始人之一Steve Wozniak一同创业，但始终不景气，她的辉煌时代已经过去。

Gil Amelio总结他在Apple时期的工作时说：“Apple是一艘底部有洞漏水的船，而我的工作是把这船引向正道。”（Apple is like a ship with a hole in a bottom, leaking water, and my job is to get this ship pointed in the right direction.）Ellen Hancock虽然同Gil Amelio一样，不知如何去堵这个漏水的洞，但正是由于她在Apple的出色表现，不但把船引到了正道上，还找来了有能力堵这个洞的人。P



王越

清华大学建筑学院毕业，现居美国。中国著名TeX开发者，非著名OpenFOAM开发者。

责任编辑：俞丽佳 (yulj@csdn.net)

# 网状数据库之父

## ——Charles Bachman

文 / 苏椰

今天我们要介绍的人物，出身于二战，他来自美国陆军防空高炮师，名叫Charles Bachman。战争结束后，Bachman进入密歇根大学，1948年获得机械工程学士学位，1950年获得宾夕法尼亚大学硕士学位，同年获得沃顿商学院MBA学位。随后他加入密歇根州的陶氏化工。1960年，Bachman跳槽加入通用电气，并在这里开发了第一代网状数据库管理系统——IDS。

1940年代，计算机在特殊的世界局势中诞生，主要为了弹道计算、密码破译和原子弹研究等工作，这些全是基于数值的。当时的计算机只能处理数值，无法处理字母等其他符号；当时也没有大容量的存储器。到了1950年代初，这两个问题得到了历史性的突破，计算机可以处理字母等各种符号了，人们把磁带系统引入了计算机，作为外部存储器。但是，磁带不仅速度奇慢，更大的问题是，它是顺序读写的，对于一条磁带而言，数据结构没有意义，想取个数是需要倒带的。1956年，IBM公司推出了磁盘系统，容量大、转速很快，几乎是随机读写的。这个质的飞跃，使得数据的逻辑结构从此是非线性的了。此时最直观的想法，自然是树状结构，IBM 360系列引入了世界上最早的数据库管理系统之一IMS，此系统为1969年的阿波罗登月立下了汗马功劳，它就是基于树状结构的。无论程序还是数据，本质都是对现实世界的一种抽象，我们希望这种抽象能尽可能贴切

反映现实世界中的关系。很明显，现实世界中的事物不都是树状的。于是我们又想到了网状结构。至此，我们就来到了Bachman当时所处的背景。

网状数据库，听起来很简单。线性结构就是树状的一种特例，我们花了10多年才完成这种泛化，因为这里面需要随机存储器，需要硬件支持。但是，从树状到网状的泛化，并不需要什么新型硬件，似乎只需要一个新型的数据库管理系统。然而这个管理系统并不容易实现。在树状结构中，任何结点有且只有一条访问路径，就像在日常操作系统中，不可能建出另外一个根目录，也不能建出一个同时属于多个目录的子目录。但在网状结构中，这都是可能的，可以有多个结点没有父结点（但此时已不能把它们叫做“根”了），一个结点也可以有多个父结点。有多个父结点就意味着，要访问一个结点，可能存在多条不同的路径。这样复杂的系统要如何管理？网状数据库还有很多类似的困难，等着人们来解决。

这一等，又将近10年。Bachman加入通用电气之后，负责涉及全公司各部门的综合管理系统，包括生产规划、配件和原材料供给、订单处理及反馈等许多方面。他要使用自家生产的GE 225计算机，设计全方位的企业生产信息和控制系统（MIACS）。1964年，该系统问世，包括很多组件，底层是集成数据存储系统，也就是我们前面所说的IDS，这就是世界上



第一款网状数据库管理系统。

时势造英雄，Bachman抓住历史时机，结合工业界的需求，创造了革命性的产品。1970年，通用电气的计算机业务被霍尼韦尔收购，Bachman来到波士顿，在霍尼韦尔高级研究部从事数据库工作。他还曾为ISO委员会设计过开放系统互连（OSI）模型，并担任过美国国家标准委员会的数据库管理系统研究组副主席，致力于数据库管理语言的标准化工作。1971年，Bachman的研究组提出了DBTG报告，其中确立了包含外部、抽象和内部的三层模式，该模式在数据库领域有着极为深远的影响力。1973年，ACM授予Bachman图灵奖，表彰他在数据库领域的杰出贡献。Bachman始终活跃在工业界，他是仅有的几位没有学术背景、没有教职、甚至没有博士学位的图灵奖得主之一。P





高巍

安卓爱普公司创始人。原搜狐媒体产品中心经理。关注移动互联网、互联网产品管理。  
微博：<http://t.sina.com.cn/inetpm>

# 新泡沫,新规则

2011年,北京的空气里充斥着一股挑逗、不安与蠢蠢欲动的气息。

如果说Facebook、Twitter、Groupon和Zynga令人眩目的高估值还只是让“泡沫”成为隐忧和争论的话题,当Color尚未发布产品,就拿到4100万美元投资时,所有人都意识到,“泡沫”真切地成为事实。

在中国,互联网公司高管离职创业的消息接踵而来:盛大在线副总裁裁边江、盛大边锋总裁许朝军、酷讯创始人陈华、多玩副总裁林军、凡客诚品助理总裁许晓辉等。突然之间,同事朋友都开始热烈讨论离职创业了。百姓网CEO王建硕开始欢呼,“2011年注定是中国互联网第三春。”

历史仿佛在重演,无法不让人回想起2000年,那场Dot-Com狂飙突进的泡沫期。

《创业家》杂志社长牛文文开始提醒创业者要“准备棉衣”,“现在无疑是创业的高峰期,但却不一定是创业的最好时机。”

商业不息,投机不止。难以想象完全摒弃投机和泡沫的商业世界。历史一再重复就必定存在规律,不能简单贴个“泡沫”标签,然后畏之如虎,而应审视这次新泡沫与10年前的不同,勇敢驾驭潮流。

首先,这一轮泡沫在云计算、开源软件、“定制”硬件与民主的新技术革命的背景下兴起。“计算”从未如此便宜、方便与简单。20世纪末那场泡沫破灭时,印象最深刻的应该是创业公司破产后,堆积如山、价值不菲的Sun服务器和EMC存储在等待拍卖。而这轮泡沫期,创业需要的启动成本,特别是硬件的初始投入,可以压得很低。比如,借鉴Quora的知识问答网站“知乎”,最早架构在Linode的VPS上,按月支付租金且极其低廉。随着存储成本不断下降,有人甚至说“读图时代”会提前到来。


其次,这轮泡沫所面对的投资环境也

大为不同。10年前的泡沫期,主要依靠传统VC, SharesPost和SecondMarket这样的二级市场模式尚未出现。现在,上一轮成功的互联网创业者如雷军、周鸿祎、曾李青等,开始成为懂行的天使投资人,类似“创新工场”这样的超级天使、天使基金不断涌现。另外,创业孵化器,甚至类似“新单位”、“车库咖啡”的虚拟办公室模式也为创业者提供了便利。更重要的,投资人的心态也逐渐成熟。IDG熊晓鸽说,“做投资,第一,不要怕泡沫;第二,有泡沫是好事。”

2000年,中国网民仅两三千万,互联网对实体经济的影响还微乎其微;如今国内互联网用户达到5亿,网络已大规模侵入众人的生活,渗透到实体经济的各个角落,信用、支付、物流等基础设施基本完善,巨大的用户规模和成熟市场更是与十年前有着本质区别。

上一轮泡沫中,富有冒险精神和表现欲的年轻人,一脸满不在乎地开公司、上市,享受眩目的财富和荣耀。今天则更多强调商业基本面、重视用户体验,小心翼翼地控制成本,如李彦宏告诫的“闹中取静”、低调务实。对大多数创业者,泡沫意味着融资机会相对增多,但不能取代孜孜不倦的耕耘。

任何一场变革中,有盲从,有投机,但即使是山寨创业,尽管创造力相形见绌,以令人吃惊的速度对世界潮流作出反馈、追随并复制,却也是几十年来罕见的。泡沫也许只催生了一群蹩脚的青年创业者,但亲历历史的骄傲感,接受技术、创新、全球化的启蒙,都将在日后发挥深远影响。

伟大的戏剧拉开帷幕了。对年轻人来说,泡沫让世界如此美好,历史似乎要重新洗牌了。正如电影《社交网络》所宣扬的,“为了从前人手中接管这个世界而全力以赴”吧! 



刘龙龙  
台湾铭传大学国际学院教授

# 软件架构与程序员的关系

软件架构大多仅是概念，以说明大型软件中主要模块区分以及各模块的关联性；这些概念的实现要靠各种接口、协议以及软件支持工具来达成。其重点之一是针对“大型软件”的开发，需由很多程序员共同完成，如果不能将其适当区分、切割，不但不易分工，最后把部分编程结果集成时，还可能是项大工程。程序员在工作时应该了解自己负责的部分在哪个模块里、如何规范与他人的接口，才能顺利完成整个项目。

早期软件架构将数据与指令分开，编程时要区分变量声明与描述，不能混同；还有函数库的应用，只要写明函数的名称就可调用，不必费心考虑如求sin函数值或printf的细节。到了1980年代，软件设计开始有主从式架构的概念，将程序分割开，按客户端/服务器两个模块编程，软件应用的灵活度大幅提升，譬如，把它们分别安装在两台计算机上，通过网络来运算。这个概念的延伸即多层（n-tier）架构，浏览器、网站服务器、应用服务器和数据库服务器等各层次分开的模块共同完成应用软件的功能。

从使用需求与厂商提供的工具来看，现在的通用软件架构大致可以区分成八个主要模块：输入、程序处理、输出、用户接口、异常处理、数据库、网络以及多地区语言（国际化与本地化）支持。各模块内部又有各自的架构，例如最前面三个模块是传统的IPO架构，用户接口模块内部大多是图形呈现以及鼠标、手势识别等可视化编程的架构；而异常处理模块内部则常采用try-throw-catch的架构。数据库模块大多以能连接关系型数据库接口为设计基础，网络模块则以能连接互联网、支持相关通信协议的接口为诉求。多地区语言支持模块则要求把程序中所有与语言文化有关的数据（包括文字与图案）独立出来、分别处理。

早期以功能为主要架构设计考虑因素的技术大多会形成树状架构；而考虑对象因素的架构设计则会形成网状架构。当代以服务为主要架构设计考虑因素的技术形成的软件架构，可以举例似乎最简单不过的SOA和不容易马上理解的REST来感受。SOA说明网络上的服务要由用户、服务查询和服务提供三个角色完成才有效率；REST说明了客户端与服务器互动时状态（state）的概念，除了每次的问答（request-response）都必须独立（stateless），还要考虑客户端处于何种状态（at rest或in transition），譬如，at rest状态才可以接受用户下一个指令。另外，如果服务器上的某个资源内容没有任何状态改变，客户端前一次取得的该资源内容则可以继续使用（如网页的cache）。

思考以上的描述，年轻的Android程序员大概能知道为何Android Development Kit环境中，系统自动有res/layout/value等文件夹的区分以及怎样运用才是有效编程。Java程序员能了解session bean与entity bean在MVC架构上的使用方式；而.NET程序员也可以回想被要求采用Code-Behind架构的背景。再往较早时期看一下，当初开发微软Windows应用软件使用SDK时，程序员就被训练成先把所有要出现在屏幕上的文字从各个程序中独立出来成为resource文档；1980年代IBM SNA/CICS程序员就采取如下的方式编程：把3270工作站上要显示的信息及可能输入的文字部分另建立成map。

现代的软件架构设计技术，都是前辈程序员靠创新、实践、经验传授等累积而得的精华，而各种搭配的软件开发工具与IDE也都非常完整。如果年轻程序员的编程工作没有采用这些工具或环境，或是虽然有采用，却不能了解它们的真正意义而误用，那可能还是土法炼钢而无法保持竞争力。P

# 10年前

## ——《程序员》2001年第5期

文 / 常政

2001年《程序员》5期重点报道的产品是Windows XP。如封面图显示，它的位置在众多标题里并不起眼。尽管历史证明这是一款经典的成功之作，不过在2001年，一切都是未知，当时Bill Gates宣布：“这将是自Windows 95 问世以来，最重要的一种Windows操作系统，它将再次激发市场对PC的需求。”但业界不以为意，毕竟微软每推一款新品总习惯赋予它史诗般的意义，而结果往往是昙花一现。比如之前的Windows 98、Windows ME，以及之后的Windows Vista。

不过这次是例外，Windows XP成为微软史上最成功的操作系统。如果包括盗版用户，自上市以来，Windows XP在全球的用户高达10多个亿。它为什么会如此成功？重温《程序员》的专题——Windows XP与开发者，我们发现，时间如流沙，近1.5万字的报道中，下面一小段文字，在历史面前越来越焕发出璀璨的金色光芒：

“Windows XP最大的变化与其是它漂亮的外观，还不如是它的设计思想：因为基于Windows 2000的NT内核，所以Windows XP彻底抛弃了古老的DOS；也因此Windows XP完全走出了个人电脑（PC）操作系统的概念，彻底以网络为基础，因为基于NT内核的Windows 2000，天然便是网络操作系统，只不过，Windows XP走得更远，它面向的是Internet。”

没错，Windows XP的成功不仅是用户体验，更在于它迎合了一个大时代，2001—2010年正是全球互联网应用、互联网用户爆发式成长的时期。当然它在中国的盛行，也得益于盗版的“东风”。

令人哭笑不得的是，由于Windows XP太受用户欢迎了，反而使得它逐渐成为微软的心中之痛，理由不难理解——商业利益，如果用户一直使用Windows XP不愿意升级换代的话，微软就无法从中获利。为了终结Windows XP，2007年1月30日微软公司祭起了它的核武器——Windows Vista。至于结局已经众所周知，微软尽管对Windows Vista用尽溢美之词，但用户并不买账，甚至“声名狼藉”：漏洞多、耗资源、兼容性差等，理由不一而足。李开复在其自传《世界因你而不同》中披露了他所亲历的Windows Vista研发细节，从中可一窥它失败的原因。据李开复回忆，Bill Gates 为Vista团队提出了3大目标：一，操作系统全用C#写；二，开发可将每一个文件存成数据库的WinFS；三，开发使得浏览器显示和传统应用相同的显示系统。这里面蕴含的技术瓶颈几乎达到了极限。而结局，李开复说：“当核心团队看到任务无法完成时，他们不再努力工作，只想看如何推卸责任。当时，几乎每一个团队都沉浸在这样的氛围里。Windows 95与Windows XP全球震撼登场的场景似乎



已经飘然而去，而成功的渴望已经变成了对失败的恐惧和对项目的怀疑。”

绞杀Windows XP的转机出现在2009年——Windows 7的横空出世。它其实是Windows Vista的升级版，但在性能、易用性、兼容性上作了大量改进，有效地整合了Windows Vista和Windows XP的优势，实现了技术和用户体验的和谐，因此一经推出，立刻受到用户们的交口称赞。据外媒报道，2011年4月，Windows 7在美国的市场份额首次超过了XP，达到31.8%，XP的份额为31.55%。看来，Windows XP的消失将是一个时间问题了，而它命运的跌宕起伏也昭示了向用户直接收费的传统商业软件商的宿命：你永远不能将产品做得太完美。P





**Robbin:**   
Unix程序员, AppleFans, Geek, 急需女朋友。命令行狂热者。

**Hank:**   
Windows程序员, .NET高手, 已婚。保守派。

**Ada:**   
架构师, 第三性别, 喜欢挖苦男人, 热爱无所不能的emacs。

**作者介绍: 西乔**  
项目经理, 06年起携创业团队从事Web技术外包开发及产品咨询顾问。



如果你有什么好玩的关于程序员的故事、对话、代码, 愿意通过漫画的形式分享, 请给西乔发邮件: arthur369@gmail.com。

## 幽默

一程序员好不容易要到美眉电话, 发短信“Hello world”, 没回; 再发“test”, 没有反应, 只好发“ping”。

一程序员去面试

面试官: 你毕业才两年, 这三年经验是哪儿来的。

程序员: 加班。

iPhone用户三大爱好: 游戏拍照发微博。

Android用户三大爱好: 刷机重启换电池。

Symbian用户三大爱好: 摔手机摔手机摔手机。

山寨用户三大爱好: 凤凰传奇凤凰传奇凤凰传奇。

程序员最大的愿望就是生三个娃儿, 一个叫Ctrl, 另一个叫Alt, 再一个叫Delete, 孩子闹腾时, 同时按一下, 世界就清静了……

“shift”是一种态度, “delete”是一种方式, 而“shift+delete”是一种勇气!

有一种崩溃叫做密码输入有误, 有一种惊慌叫做账号异地登录, 有一种感情叫做隐身对其可见, 有一种误会叫做人机离线, 有一种失落叫做你没有访问权限。

Ruby程序语言里p表示打印, 给同事做代码审查, 有个接口写了很多打印语句, 于是斩钉截铁地跟他说: “你放的p太多了!”

做为一名程序员, 每次看到工资单的时候, 我都会想: 那些数字为什么不是16进制!

男程序员: 我是左括号, 你是右括号, 我们不在一起就会编译错误。

女程序员: 但我是小括号, 你是花括号, 就算在一起, 也还是编译错误。